

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-210686

(43) 公開日 平成7年(1995)8月11日

(51) Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T 7/00				
G 0 3 B 35/00	Z			
G 0 6 T 7/20				
		9061-5L	G 0 6 F 15/ 62 15/ 70	4 1 5 4 1 0
審査請求 未請求 請求項の数2 F D (全106頁) 最終頁に続く				

(21) 出願番号 特願平6-282980

(22) 出願日 平成6年(1994)10月21日

(31) 優先権主張番号 1 4 1 1 5 7

(32) 優先日 1993年10月21日

(33) 優先権主張国 米国 (U S)

(71) 出願人 591264544

イーストマン・コダック・カンパニー  
アメリカ合衆国、ニュー・ヨーク・14650、  
ロチェスター、ステイト・ストリート・  
343

(72) 発明者 サージェイ ブイ フォーゲル

アメリカ合衆国 ニューヨーク州 ロチェ  
スター イースト アベニュー 1577

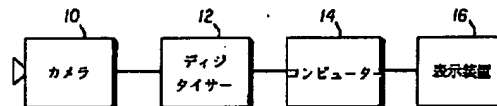
(74) 代理人 弁理士 吉田 研二 (外2名)

(54) 【発明の名称】 ステレオ画像から奥行き画像のための中間画像を構成する方法および装置

(57) 【要約】

【目的】 本発明は、複数の中間画像を作成するために一対のステレオ画像間に補間操作を実行する。

【構成】 この補間操作は前記中間画像の速度ベクトル場を評価することを含む。ある特定の中間画像のための評価は、システムがより最適な解に到達できるように前記二つの画像間の対応の探索を水平方向に制約することを含む。平滑化による過度の間隙充填は、不必要である。同時に前記プロセスは、前記画像全体を垂直方向に整列させる。



## 【特許請求の範囲】

【請求項1】 (a) 水平方向に別れている一つのシーン内の一対のステレオ画像を捕らえるステップと、

(b) 水平方向に制約された速度ベクトルの評価によってその一対の画像間に中間画像を生成するステップとからなることを特徴とする方法。

【請求項2】 一対のステレオ画像を捕らえる撮像装置と、

前記撮像装置に接続されており、水平方向に制約された対応の補間により中間画像作成し、かつ前記画像対と前記中間画像とから前記奥行き画像を作成するコンピューターと、

前記コンピューターに接続されており、かつ前記奥行き画像を出力する奥行き画像出力装置とからなることを特徴とする奥行き画像作成装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、中間画像を内挿してそれらを一つの画像にインターレースすることにより、一対のステレオ画像から一つの奥行き画像を構成する方法およびその装置に関する。

## 【0002】

【関連技術】奥行きの知覚を有する画像を生成することは、伝統的に写真的方法で行われてきた。インテグラル・レンティキュラー写真術は理論的な考察と実証の長い歴史を持っているが、商業的には限られた成功を得ただけであった。インテグラル・レンティキュラー写真術を支持する基本概念の多くは何年も前から知られている

(Takanori Okoshi (大越孝敬) 著: Three-Dimensional Imaging Techniques (三次元画像形成技術)、Academic Press, New York, 1976; および G. Lippman 著: "Epreuves re'versibles, Photographies integrales," Comptes Rendus, 146, 446-451, March 2, 1908 を参照)。インテグラル写真術と言う用語は、多数の顕微鏡的に微細な写真画像構成要素の集積として全体画像の構成を行うことを言っている。各々の写真画像構成要素は通常、同一の浮彫り加工された複数の球面からなるモザイクの一部として形成された別々の小さなレンズ、さもなければ適当な厚さのプラスチックシートの前面に形成された別々の小さなレンズを通して眺められる。このシートはその後、写真画像構成要素を含むエマルジョン層と密着して接合または保持される。レンティキュラー写真術は、インテグラル写真術の特殊な場合と考えられる。この場合は複数の小さなレンズがプリント領域の全面を垂直方向に走る複数の円筒切片に形成されている。レンティキュラー写真術の形式での最近の商業的試みには、現在、香港のカメラ工場が製造している Nishika カメラとして販売されている Nimslo カメラがある。奥行きの感じは明瞭に目視できるが、得られた画像には限定された奥行き感があり、プリントを揺らしたり

そのプリントに対する目視者の視点が変化すると画像がジャンプするように見える。

【0003】レンティキュラー写真を作成する光学的方法は Okoshi (大越) によって述べられている。所望のシーンの方向に対して直角な水平方向に平行移動を可能とするスライドレール上のキャリッジに写真カメラを取り付ける。中心の視点からその中心の視点の両側の視点に等間隔で露出を行ってカメラを移動させながら一連の写真を撮影する。これら横方向の視点と中心の視点との距離は、レンティキュラー材料が任意の所定のレンティキュール(微細レンズ)に隣接するレンティキュールの背後にある写真画像を投影し始める前に、その所定のレンティキュールの背後にある写真画像を投影することができる最大角度に依存する。中心の視点からの写真を含むことは必要でなく、その場合は画像の数は偶数になる。中心の視点からの写真を含む場合は、画像の数は奇数になる。横方向の視点を含んだ視点間に含まれるビューの全数の合計が最終的に各レンティキュールの背後に含まれる写真構成要素の数を決定する。

【0004】これらのビューの各々から結果として得られるネガ画像は、そのカメラレンズと同じ焦点距離のレンズを装備した引き伸ばし器に入れられる。このカメラは前述のように、逐次行われる露出と露出の間に横方向に移動されるから、元のシーンの複数の画像の位置はフィルム枠を横切って横方向に移動する様に見える。結局、逐次的ネガ画像がフィルムゲートに置かれるから、これらのネガから引き伸ばされた画像の位置もまた引き伸ばし器のイーゼルの中心に関して水平方向に移動する。片側だけ浮彫り加工されたレンティキュールを有する適当な厚さの透明プラスチックシートの平らな裏面に接しかつそのエマルジョンで方向づけされた一枚の写真材料からなるアサンプラージ(写真の集まり)が、そのレンティキュラー側を引伸ばしレンズに向けて引伸ばし器のイーゼルの上に置かれる。イーゼル上のこのシートの位置は中心の画像のフィールドがこのアサンプラージの中心に来るまで調整され、そして引伸ばしレンズから投影されている情報の露出はこれらのレンティキュールを通して感光乳剤上に行われる。それから、逐次行われた露出から得られたネガ画像がフィルムゲートに置かれ、このアサンプラージの位置がそれぞれのビューをアサンプラージの中心に置きなおす様にイーゼル上で再調整され、そして引伸ばしレンズから投影されている情報の追加露出が行われる。横方向の視点間に含まれる全てのビューが同様にしてレンティキュラー・プラスチックシートを通してエマルジョン上に露光されると、このフィルムシートはレンティキュラー・プラスチックシートから分離して現像することができる。もし引伸ばしレンズのアパーチャ径が交互のビュー間の横方向のシフト量に等しくセットされているならば、各レンティキュールの背後の空間はちょうど一杯に写真画像構成要素で満たされ

ることが分かる。このプロセスの最終段階は、エマルジョン層とレンティキュラー・プラスチックシートの平らな面との間で密着するように写真フィルムとプラスチックシートとを再組み立てすることであり、また円筒形のレンティキュールを通した露光の結果得られた隣り合った画像の長いストリップが再び観察用のレンティキュールの下で同様な仕方では位置決めされるように、横方向の位置決めをすることである。

【0005】理想的にはインテグラル写真あるいはレンティキュラー写真は各小レンズあるいは各レンティキュールから無限の異なった角度のビューを表示する。各角度のビューは対応する有限の小面積の露光エマルジョンまたは他のハードコピー媒体を持たなくてはならないから、これは不可能である。したがって上限として、ビューの数はハードコピー媒体の解像度限界を超えてはならない。Nimsloプリントでは、各レンズの背後のビューの数は4個に限定されており、その内の2個は左の透視画ビュー、残りの2個は右の透視画ビューと考えられた。これは十分に通常の写真乳剤の解像度限界以下であって、観察者の頭が横方向に動く時の立体視透視画のオブションを二つだけ見込んでいる。

【0006】光学的多重投影法はまた、1960年台と1970年台にEastman Kodak社の研究者と技術者達によって多くの実験でも利用され、また1969年の株への年次報告書の表紙を飾ったレンティキュラー写真を生み出した。このプリントは各レンズ毎に透視画の移り変わりがスムーズになるように、シーンの交互のビューを取るカメラを数多く使った。21個もの異なる角度のビューを表示でき、結果ははるかに効果的であった。この画像記録法は最終的なプリント記録が一連の二次元画像記録から間接的に導き出されるので「間接」技法と呼ばれる。レンティキュラープリントあるいは奥行き画像を生成するさらに近代的な方法は、同一シーンの空間的に分離された二つ以上の画像を写真的に捕らえてそれらの画像をデジタル化するか、デジタル画像を電子的に捕らえることである。これらのデジタル化された画像は、観察用にレンティキュラー・フェースプレートが取り付けられるプリント媒体あるいは透明媒体上に電子的に画像をプリントするようにプリンターを駆動するために使われる。この奥行き画像生成法は米国特許5, 113, 213号と米国出願一連番号No. 885, 699 (Kodak Dk l. 64, 374)とによって代表される。

【0007】

【発明が解決しようとする課題】上記の方法の一つの問題は、画像の撮影はプリントする場所から離れたところで行われることが非常に多いということである。もし写実的な奥行き感を出すためにそのシーンのさらに多くの画像が必要であるならば、撮影等のプロセスが再び完全に行われなくてはならない。画像はデジタルであってコンピューターによるプリント用に処理されるから、コ

ンピューターを利用して補間法を用いて余分に必要な画像を生成することにより、この問題を解決することができる。しかしながら、対象遮蔽に関連する問題のために通常の補間法では適当でない。余分に必要な画像を生成するためのプロセスについては、米国出願07/722, 713号に記述されている。しかしながらこの手法は、多次元探索が行われているときの多次元探索と画像平滑化の要件とのために、中間画像の特徴探索の間に必ずしも最適解が得られないという理由で、最善の中間画像を生成できない。

【0008】上記システムの第二の問題は、補間される中間画像を生成するためにその中間画像を挟む複数の画像を記憶しておかなければならないということである。

【0009】必要とされるのは、あまり画像平滑化を必要としない探索に対する一層最適な解が得られるという理由で中間画像の品質が良いというような一対の撮影された画像の間に多数の中間画像を生成するステレオ画像補間システムである。

【0010】

【課題を解決するための手段及び作用】本発明は、見る人に一層魅力的な奥行き画像を提供することを目的とする。

【0011】本発明の他の目的は、一対のステレオ画像から数個の中間画像を生成することである。

【0012】本発明はまた、補間された画像の品質を改善するためおよび画像平滑化処理による画像間隙充填のための必要性を減らすために画像探索処理を水平方向に制約することを目的とする。

【0013】本発明はさらにまた、一回の操作で複数のステレオ画像全体を垂直方向に整列することを目的とする。

【0014】本発明は、一つの奥行き画像を生成するために使われる複数の中間画像を生成するために一対のステレオ画像間で補間操作を行うことによって上記の目的を達成する。この補間操作はこれら中間画像の速度ベクトル場の評価を含む。ある特定の中間画像についての評価は、二つの画像間の対応性の探索を水平方向に制約することを含む。このことは、システムが、結果的に平滑化による過度の間隙充填を必要とせずまた同時に全画像を垂直方向に整列する一層最適な解に到達するのを可能にする。

【0015】後に明らかになる他の目的や利点と共にこれらのことは、以下にさらに十分に記述し請求するような構成と操作の詳細の中に含まれており、また本文書の一部をなす添付図面を参照している。これらの図面全体を通して同じ数字は同じ部分を指している。

【0016】

【実施例】本発明は、レンティキュラー画像あるいはバリアー画像といった奥行き画像が生成できるように一対のステレオ画像に対して複数の中間画像を得ることを意

図している。図42に示す如く、本発明は、通常の画像処理装置を使って実現される。この装置は、フィルム上にステレオ画像を撮影する1個以上のカメラ10を含んでいる。フィルム画像は、通常のデジタイザ12によってデジタル化される。勿論、フィルムカメラとデジタイザの代わりに電子カメラを使うこともできる。デジタイザ画像は、ここで論じられる処理を実行して中間画像を生成するコンピューター14に供給される。これらの画像は、インターレースされて表示装置16に与えられる。この表示装置は、レンティキュラー・フェースプレートまたはバリアーストリップ・フェースプレートを有するCRTであってもよく、またあるいはレンティキュラー・フェースプレートまたはバリアーストリップ・フェースプレートが接着されたプリントを作成するプリント・プリンターまたは透明フィルム・プリンターであってもよい。もっとも魅力的な奥行き感を得るためには、最多数の角度ビューによって奥行き画像を生成することが望ましい。

【0017】また光学的に記録された画像の数を一対のステレオ画像にまで減らすこと、および欠落している角度ビューに対応する画像（中間画像）をこのステレオ対から再構築することも望まれる。中間画像を再構築するためには、まずステレオ対の間の対応を確定し、それからこれらの対応を使って補間処理を通して中間画像の評価を行う。ステレオ対間の対応は、カメラレンズの位置の違いによって生ずる二つの画像の見かけの動きを確定する。ある点の位置の変化を一つのベクトルと見なすことにより、あたかも二つの現実のカメラの中間に一つのカメラが置かれているかのように画像を評価する手段が確立される。主として遮蔽によって必ずしもすべての点30が対応点を持つわけではないので、これらの発生に対して画像を再構築するための許容値を設けなければならない。このベクトル場は一般に光学流れと呼ばれる（Robot Vision, B.K.P. Horn, The MIT Press, 1991）。光学流れのよく知られた特徴付けの一つは、奥行きは光学流れベクトルの大きさに反比例するということである。この場合、光学流れベクトルの大きさがゼロの物体空間に、すなわち背景のようにカメラ位置の変化に影響されない点にゼロパララックス平面が発生する。

【0018】中間画像22の画素20（図43を参照）40を生成するに際して、第一のステップは一対の画像24と26の間の対応を見つけることである。画像対の対応部分を見つけるためには、中間画像の中に生成される画素20を通るベクトルあるいは線28を用いて画像24、26上の比較すべき異なった領域を指定する。本来、画像24、26上の結合点は、スペクトルその他の画像特性においてもっとも高い相関性を持った領域を探索しながら、二次元領域30、32のあたりを系統的に移動できる。一旦これらの点が見つかれば、その対応を使って中間画素20の値を補間することができる。二次

元での2画像間の対応は一意的解を持たないから、結果的に図43に示すような探索処理は最適解を与えないことがある。さらに、解が一意的でないで中間画像22の中に生成された雑音成分を削減するか除去するために画像の平滑化を行うことが必要である。

【0019】ステレオ画像が関連するときに対応を確立するプロセスは、これら二つの画像が同一のシーンのものであると言うことを認識することによって強化でき、またその解は改善できる。唯一の違いは、画像がそこから撮影されたビューあるいは点の水平方向の変位である。図43にあるような二次元領域の探索の代わりに、図44に示すようにこれらの画像の水平軸38に沿った一次元探索が行われる。本発明の概念のこうした理解が得られれば、本発明によって解決される問題はより詳細に述べることができる。

【0020】一対のステレオ画像（一つのシーンを異なる視点から撮影した画像）が与えられれば、中間画像（いくつかの中間視点から撮影された画像）の評価は見いだされるに違いなく、またその奥行き画像はこれらの中間画像をインターレースすることにより構成されるに違いない。中間画像を評価するためには、ステレオ画像対間の対応を確立し、それからこれらの対応を使って補間処理を通して中間画像を評価しなくてはならない。シーン座標系の座標軸は水平軸、垂直軸、および奥行き軸と定義される。視点の選択は、ステレオ画像対の画像平面が水平軸と垂直軸とを含むシーン平面と一致するような仕方限定される。ステレオ画像対の撮影に使われたカメラは水平軸と垂直軸とを通るシーン平面に平行に置かれておりと仮定されており、またカメラ位置の変化は厳密に水平方向であると仮定されている。またステレオ画像対の各画像の水平軸と垂直軸はシーンの水平軸と垂直軸と一致すると仮定している。これらの仮定の下で、ステレオ画像対の画像間の対応は、画像の整列に関する一定の垂直成分とシーン内の対応点の奥行きに関する可変の水平成分を持つ中間ディスプレイ・ベクトル場によって特徴付けることができる。

【0021】以下に論ずるようこの問題を解決するために、あるステレオ画像対を形成するそして一つのカメラで撮影されたシーンの左右の画像の全体像は固定数のスペクトル・バンドBを持った多重バンド・デジタル画像にデジタル化される。左右のデジタル画像の各バンドは、感知された画像の上のグリッドの点 $\Omega$ から取られた、画素として知られる幾つかの測定値を表す数の二次元配列として定義される。本好適な実施例においては、このグリッドは矩形の格子であると仮定している。視点 $t_r$ で撮影され、スペクトルバンド $R_1, \dots, R_n$ からなる右デジタル画像50（図45を参照）が与えられ、また視点 $t_l$ で撮影され、スペクトルバンド $L_1, \dots, L_n$ からなる左デジタル画像52が与えられると、バンド $D_1, \dots, D_n$ からなる

奥行き画像を構成するプロセスは図45に図解され、また次のように説明できる。まず初めに、左右の画像間の対応が確定され54、そして幾つかの中間視点 $t_1, \dots, t_k$ における中間ディスパリティ・ベクトル場 $(U_1, V_1), \dots, (U_k, V_k)$ が評価される56および58。それから各バンド $b=1, \dots, B$ ごとに、また各中間視点 $t_k, k=1, \dots, K$ ごとに中間画像バンド $I_{b,k}$ の評価が補間処理によって得られる60。最後に各画像バンドごとに奥行き画像バンド $D_b$ は、奥行き画像を生成するために中間画像バンド $I_{b,1}, \dots, I_{b,k}$ をインターレースすることにより構成される62。各 $k=1, \dots, k$ に関して中間視点 $t_k$ における中間ディスパリティ・ベクトル場 $(U_k, V_k)$ を評価する方法は図46に図解されており、また下記の多段レベル解像度処理として説明することができる。すなわち、1) もっとも粗い解像度レベルで中間ディスパリティ・ベクトル場の現行評価から始める70、(最初にある定数に設定する)；2) 現行評価を繰り返し改善する72~82；3) 中間ディスパリティ・ベクトル場の現行評価と見なされる次段のより細かいレベルでの中間ディスパリティ・ベクトル場の初期評価を得るために次段のより細かい解像度レベルに投影する84；4) もっとも細かい解像度レベルに達するまでステップ2)とステップ3)を繰り返す。

【0022】最高解像度レベルの適切な評価は、中間ディスパリティ・ベクトル場の最終評価と見なされる。

【0023】多段レベル解像度処理の各レベルでは、中間ディスパリティ・ベクトル場の未知の現行評価に関する連立非線形方程式が形成され76、それからこの連立非線形方程式を解く78ことにより中間ディスパリティ・ベクトル場の現行評価の反復改善が達成される。中間ディスパリティ・ベクトル場の未知の現行評価に関する連立非線形方程式は、中間ディスパリティ・ベクトル場の未知の現行評価と、左右の画像に関する数量と、左右の画像の一般空間偏導関数とによって定義される。これらの数量は、フィルター $\phi_1, \dots, \phi_r$ とこれらのフィルターの空間偏導関数とで左右の画像の各バンドをフィルタリングすることによって得られる。記号 $G$ は、ある特定のバンドとある特定のフィルターとを指定する各インデックス $g \in G$ を持ったインデックスの集合のために使われる。

【0024】多段レベル解像度処理(本書の参考資料に入っている米国出願631,750を参照のこと)は、各々のより粗い解像度レベルにおいてフィルタリングされた左右の画像とそれらの空間偏導関数と中間ディスパリティ・ベクトル場の評価とが、より細かい解像度レベルのグリッドのサブグリッド上で定義されるということの意味する多段レベル解像度ピラミッドのまわりで構築される。

【0025】所定の解像度レベルで、フィルタリングさ

れた右画像のバンドとフィルタリングされた右画像のバンドの空間偏導関数とフィルタリングされた左画像のバンドとフィルタリングされた左画像のバンドの空間偏導関数とは、サイズ $N' \times M'$ の画像平面上の点 $Q'$ の同一グリッド上で定義される。同じ解像度レベルで、中間ディスパリティ・ベクトル場は、サイズ $N \times M$ の画像平面上の点 $Q$ のグリッド上で定義される(以下の論議では、垂直方向に関して取られた空間偏導関数は下付き文字 $v$ を持ち、同様に水平方向に関して取られた空間偏導関数は下付き文字 $u$ を持つ)。この解像度レベルで連立非線形方程式を形成するには次のように進める。すなわち：ある特定のフィルターとある特定の画像バンドとを指定する各インデックス $g \in G$ に関して、評価後の右のフィルタリングされた画像バンドと評価後の左のフィルタリングされた画像バンドとの間の差としてグリッド $Q$ に関する光学流れ関数 $g_u$ を形成する。評価後のフィルタリングされた右画像バンドを得るために、シーン内の可視点の画像平面への、視点 $t_k$ から取られた評価後の透視図的投影である画像点のグリッド $Q_k$ を求める。但し、視点 $t_k$ から画像平面上へのその透視図的投影はグリッド点 $Q$ である。それからフィルタリングされた右画像が定義されている点のグリッド $Q'$ から点のグリッド $Q_k$ にそのフィルタリングされた右画像を補間する。同様にして、評価後のフィルタリングされた左画像バンドが決定される。画像点のグリッド $Q_k$ を得るために、グリッド $Q$ の各画像点は、そのグリッド点で定義された評価後の中間ディスパリティベクトルの、因子 $(t_k - t_1) / (t_k - t_1)$ によって決められる値に等しい量だけシフトされる。画像点のグリッド $Q_k$ を得るために、グリッド $Q$ の各画像点は、そのグリッド点で定義された評価後の中間ディスパリティベクトルの、因子 $(t_k - t_1) / (t_k - t_1)$ によって決められる値に等しい量だけシフトされる。

【0026】ある特定のフィルターとある特定の画像バンドとを指定する各インデックス $g \in G$ ごとに、評価後のフィルタリングされた右画像の、因子 $(t_k - t_1) / (t_k - t_1)$ によって決められる空間偏導関数と評価後のフィルタリングされた左画像の、因子 $(t_k - t_1) / (t_k - t_1)$ によって決められる空間偏導関数との、グリッド $Q$ 上で定義された合計として、インデックス $g$ に対応する中間ディスパリティ・ベクトルの現行評価の成分 $u$ に関する光学流れ関数 $g_u$ の空間偏導関数 $g_{uv}$ を形成する。評価後のフィルタリングされた右画像バンドの空間偏導関数を得るために、シーン内の可視点の画像平面への、視点 $t_k$ から取られた評価後の透視図的投影である画像点のグリッド $Q_k$ を求める。但し、視点 $t_k$ から画像平面上へのその透視図的投影はグリッド点 $Q$ である。それからフィルタリングされた右画像の空間偏導関数が定義されている点のグリッド $Q'$ から点のグリッド $Q_k$ にそのフィルタリングされた右画像の空間

偏導関数を補間する。評価後のフィルタリングされた左画像バンドの空間偏導関数を得るために、シーン内の可視点の画像平面への、視点  $t_k$  から取られた評価後の透視面的投影である画像点のグリッド  $\Omega_k$  を求める。但し、視点  $t_k$  から画像平面上へのその透視面的投影はグリッド点  $\Omega$  である。それからフィルタリングされた左画像の空間偏導関数が定義されている点のグリッド  $\Omega'$  から点のグリッド  $\Omega_k$  にそのフィルタリングされた左画像の空間偏導関数を補間する。画像点のグリッド  $\Omega_k$  を得るために、グリッド  $\Omega$  の各画像点は、そのグリッド点で定義された評価後の中間ディスパリティ・ベクトルの、因子  $(t_k - t_l) / (t_k - t_l)$  によって決められる値に等しい量だけシフトされる。画像点のグリッド  $\Omega_k$  を得るために、グリッド  $\Omega$  の各画像点は、そのグリッド点で定義された評価後の中間ディスパリティ・ベクトルの、因子  $(t_k - t_l) / (t_k - t_l)$  によって決められる値に等しい量だけシフトされる。

【0027】ある特定のフィルターとある特定の画像バンドとを指定する各インデックス  $g \in G$  ごとに、インデックス  $g$  に対応する評価後のフィルタリングされた右画像バンドの画像座標系の垂直成分に関する、因子  $(t_k - t_l) / (t_k - t_l)$  によって決められる空間偏導関数とインデックス  $g$  に対応する評価後のフィルタリングされた左画像バンドの画像座標系の垂直成分に関する、因子  $(t_k - t_l) / (t_k - t_l)$  によって決められる空間偏導関数との、グリッド  $\Omega$  上で定義された合計として、インデックス  $g$  に対応する中間ディスパリティ・ベクトルの現行評価の成分  $v$  に関する光学流れ関数  $g_v$  の空間偏導関数  $g_{v,k}$  を形成する。評価後のフィルタリングされた右画像バンドの画像座標系の垂直成分に関する空間偏導関数を得るために、シーンの中の可視点の画像平面への、視点  $t_k$  から取られた評価後の透視面的投影である画像点のグリッド  $\Omega_k$  を求める。但し、視点  $t_k$  から画像平面上へのその透視面的投影はグリッド点  $\Omega$  である。それからフィルタリングされた右画像の画像座標系の垂直成分に関する空間偏導関数が定義されている点のグリッド  $\Omega'$  から点のグリッド  $\Omega_k$  にそのフィルタリングされた右画像の画像座標系の垂直成分に関する空間偏導関数を補間する。評価後のフィルタリングされた左画像バンドの画像座標系の垂直成分に関する空間偏導関数を得るために、シーンの中の可視点の画像平面への、視点  $t_k$  から取られた評価後の透視面的投影である画像点のグリッド  $\Omega_k$  を求める。但し、視点  $t_k$  から画像平面上へのその透視面的投影はグリッド点  $\Omega$  である。それからフィルタリングされた左画像の画像座標系の垂直成分に関する空間偏導関数が定義されている点のグリッド  $\Omega'$  から点のグリッド  $\Omega_k$  にそのフィルタリングされた左画像の画像座標系の垂直成分に関する空間偏導関数を補間する。画像点のグリッド  $\Omega_k$  を得るために、グリッド  $\Omega$  の各画像点は、そのグリッド点で定義された評

価後の中間ディスパリティ・ベクトルの、因子  $(t_k - t_l) / (t_k - t_l)$  によって決められる値に等しい量だけシフトされる。画像点  $\Omega_k$  のグリッドを得るために、グリッド  $\Omega$  の各画像点は、そのグリッド点で定義された評価後の中間ディスパリティ・ベクトルの、因子  $(t_k - t_l) / (t_k - t_l)$  によって決められる値に等しい量だけシフトされる。

【0028】矩形グリッド  $\Omega$  の各非境界グリッド点は、画像平面上で異なる8方向を指定する8個の最近傍点に囲まれている。矩形グリッド  $\Omega$  の各境界点に関しては、最近傍点はこれらの異なる8方向の中のいくつかについてのみ存在する。記号  $s$  は、画像平面上の異なるこれら8方向のなかの一つを指定する、画像平面上の一つのベクトルを表すために用いられる。記号  $S$  は、これら8個のベクトルの集合を表すために用いられる。画像平面上のある特定の方向を指定する、集合  $S$  の中のベクトル  $s$  のおのおのに関して、中間ディスパリティ・ベクトル場の現行評価の水平成分  $u$  の方向導関数に対する有限差分近似として中間ディスパリティ・ベクトル場の現行評価の可変水平成分  $u$  のための方向平滑化関数  $(s, \nabla u)$  を形成する。この方向平滑化関数  $(s, \nabla u)$  は、矩形サブグリッド  $\Omega_s$  のすべてのグリッド点は矩形グリッド  $\Omega$  上で方向  $s$  近傍に最近傍点を持つという性質により矩形グリッド  $\Omega$  の矩形サブグリッド  $\Omega_s$  上で定義される。矩形サブグリッド  $\Omega_s$  の各グリッド点において、この方向平滑化関数  $(s, \nabla u)$  は、このグリッド点の方向  $s$  近傍の最近傍点における中間ディスパリティ・ベクトル場の現行評価の水平成分  $u$  の値とこのグリッド点における中間ディスパリティ・ベクトル場の現行評価の水平成分  $u$  の値との差に等しい。

【0029】中間ディスパリティ・ベクトル場の未知の現行評価に関する連立非線形方程式は、インデックス  $g \in G$  によって指定される各フィルターおよび各画像バンドに関する光学流れ関数  $g_v$  およびその空間偏導関数  $g_{v,k}$  と、各画像方向  $s \in S$  に関する方向平滑化関数  $(s, \nabla u)$  と、加算、減算、乗算、除算といった四つの基本的代数演算を用いる幾つかの定数パラメーターと一緒に組み合わせることによって形成される。

【0030】右視点  $t_k$  から撮影されかつバンド  $R_1, \dots, R_B$  からなる右デジタル画像と左視点  $t_l$  から撮影されかつバンド  $L_1, \dots, L_B$  からなる左デジタル画像とを与えると、中間視点  $t_k$  で得られた中間ディスパリティ・ベクトル場  $(U_k, V_k)$  の評価に基づいて、中間視点  $t_k$  から撮影されかつバンド  $I_{1,k}, \dots, I_{B,k}$  からなる中間デジタル画像を評価する方法は、次のように説明できる。各画像バンド  $b = 1, \dots, B$  ごとに、中間デジタル画像バンド  $I_{b,k}$  の評価は、右デジタル画像バンドから予測された因子  $(t_k - t_l) / (t_k - t_l)$  によって決まる評価と左デジタル画像バンドから予測された因子  $(t_k - t_l) / (t_k - t_l)$  によって決まる評

$-t_i) / (t_i - t_i)$  によって決まる評価との和として定義される。評価後の右デジタル画像バンドを得るために、シーン内の可視点の画像平面への、視点  $t_i$  から取られた評価後の透視面的投影である画像点のグリッド  $\Omega_i$  を求める。但し、視点  $t_i$  から画像平面上へのその透視面的投影はグリッド点  $\Omega$  である。それから右デジタル画像が定義されている点のグリッド  $\Omega$  から点のグリッド  $\Omega_i$  にその右デジタル画像バンドを補間する。評価後の左デジタル画像バンドを得るために、シーン内の可視点の画像平面への、視点  $t_i$  から取られた 10 評価後の透視面的投影である画像点のグリッド  $\Omega_i$  を求める。但し、視点  $t_i$  から画像平面上へのその透視面的投影はグリッド点  $\Omega$  である。それから左デジタル画像が定義されている点のグリッド  $\Omega$  から点のグリッド  $\Omega_i$  にその左デジタル画像バンドを補間する。画像点のグリッド  $\Omega_i$  を得るためには、グリッド  $\Omega$  の各画像点 20 は、そのグリッド点で定義された評価後の中間ディスパリティ・ベクトルの、因子  $(t_i - t_i) / (t_i - t_i)$  によって決められる値に等しい量だけシフトされる。画像点のグリッド  $\Omega_i$  を得るために、グリッド  $\Omega$  の各画像点 30 は、そのグリッド点で定義された評価後の中間ディスパリティ・ベクトルの、因子  $(t_i - t_i) / (t_i - t_i)$  によって決められる値に等しい量だけシフトされる。

【0031】本発明の基本操作についてのさらに十分な理解を得るために、さらに詳細な説明を以下に述べる。

【0032】以下の説明では、より一般的な問題が考慮され、表記法の一部が変更される。視点  $t_i$  から撮影されかつバンド  $R_1, \dots, R_k$  からなる上記の右デジタル画像と視点  $t_i$  から撮影されかつバンド  $L_1, \dots, L_k$  からなる上記の左デジタル画像とは両者とも 30 輝度画像関数  $\xi(x, y, t)$ 、 $\xi \in \Xi$  によって表される。固定値  $k=1, \dots, K$  に関しては、中間視点  $t_i$  で定義された上記の中間ディスパリティ・ベクトル場  $(U_i, V_i)$  は、下記において助変数速度ベクトル場 評価----1---- によって表される。ステレオ視点の対  $\{t_i, t_i\}$  は、以下において視点列----2---- によって表される。ここで  $K=1$  である。

【0033】ある間隔  $T$  以内の視点から撮影された三次元シーンの画像とそれらの空間偏導関数とを取り扱う。 40 三次元シーンの画像の初期画像は一般に不連続関数であって、それらの空間偏導関数は不連続点では定義されない。不連続点は画像の中の最重要点になっていることが多く、容易には無視できない。これらの困難を克服するために、初期画像は一般関数として扱われ、またこれらの偏導関数は一般偏導関数として扱われる。これらの一般関数は、無限に微分可能な関数の集合のある部分集合上で定義される。この部分集合からの関数は“検定関数”と呼ばれる。それから検定関数の集合から取られた関数の特定の助変数族上の初期画像と関連付けられてい 50

る一般関数の方程式を介して二次画像の助変数族が導入される。これらの二次画像は無限に微分可能であり、これらの偏導関数は、この特定の助変数族の関数と関連付けられている一般関数を介して得られる。このプロセスは次のように説明できる。

【0034】所定の視点  $t \in R$  (ここで  $R$  は一次元ユークリッド空間である) に関する初期画像を構成する一つの方法は、シーン内の物体によって反射される光を“画像平面”と呼ばれる二次元ユークリッド空間  $R^2$  に投影し、それから画像平面  $R^2$  内の各点  $(x, y)$  の輝度値  $\xi(x, y, t)$  を識別することによって行う。上記のように定義された関数  $\xi(x, y, t)$  は“輝度画像関数”と呼ばれる。点  $(x, y) \in R^2$  と視点  $t \in R$  とにおける輝度画像関数の値  $\xi(x, y, t)$  は撮像されるシーン内のその点の輝度に略比例すると仮定されている。これは、すべての  $(x, y) \in R^2$  とすべての  $t \in R$  に関して視点  $t$  に於けるこういう点  $(x, y)$  に投影する。異なる輝度画像関数  $\xi(x, y, t)$ 、 $(x, y, t) \in R^3$  (ここで  $R^3$  は三次元ユークリッド空間 である) は、画像形成プロセスの下記の局面を変化させることにより：すなわち、シーンを照明している光源の方向と、こういう光源の色と、輝度を計算するために使われるスペクトル応答関数とを変化させることにより得られる。

【0035】各輝度画像関数  $\xi(x, y, t)$  は、 $R^3$  内の Lebesgue 測度  $dx dy dt$  に関して局所的に積分可能であり、またそのために、局所的凸型線形位相空間  $\Phi(R^3)$  上で定義される連続線形汎関数  $\Gamma$  (一般関数) を形成するために使うことが出来る。この空間  $\Phi(R^3)$  は集合  $R^3$  内のコンパクト支持を有するすべての無限に微分可能な関数からなる。これは、部分集合  $S_0$  の外側にある点では関数  $\phi$  がゼロの等しいような有界閉部分集合  $S_0 \in R^3$  が各関数  $\phi \in \Phi(R^3)$  ごとに存在するということを意味している。空間  $\Phi(R^3)$  の位相はある一定の半ノルム族によって定義される。集合  $\Phi(R^3)$  からの関数  $\phi$  は“検定関数”と呼ばれる。検定関数  $\phi \in \Phi(R^3)$  における輝度画像関数  $\xi$  と関連付けられた一般関数  $\Gamma$  の値は次の関係：----3---- によって定義される。異なる輝度画像関数  $\xi(x, y, t)$  と関連付けられた一般関数  $\Gamma$  は、族----4---- に一体化される。ここで表記を簡単にするために、記号  $\xi$  はこういう輝度画像関数のための表記法の一部としてのその役割に加えてある特定の輝度画像関数  $\xi(x, y, t)$  と関連付けられた一般関数  $\Gamma$  を指定するインデックス (助変数) の役割をする。記号  $\Xi$  はインデックスの集合を表す。

【0036】所定の視点  $t \in R$  における初期画像を構成する他の方法は、輝度画像関数によって表される投影光パターンにおける有意な変化が視点  $t$  で起こるような、画像平面内で“特徴点”と呼ばれる点----5---- の集合を識別し、それから特徴値  $\eta(x_u, y_u, t_u)$  各特

徴点——6——に割り当てることによって行われる。特徴点の集合——7——は空間 $R^3$ の閉部分集合であると仮定されている。上記のように集合 $M$ 上で定義される関数 $\eta(x_\mu, y_\mu, t_\mu)$ は“特徴画像関数”と呼ばれる。異なる特徴画像関数 $\eta(x_\mu, y_\mu, t_\mu)$ ,  $(x_\mu, y_\mu, t_\mu) \in M$ は、特徴点の集合 $M$ を選択する基準と特徴値 $\eta(x_\mu, y_\mu, t_\mu)$ を割り当てる基準とを変更することによって得られる。例えば、集合 $M$ は、空間 $R^3$ の部分集合の以下の4タイプ：三次元領域と二次元領域と一次元輪郭と孤立点との有限の組み合わせである。

【0037】集合 $M$ の“Borel 部分集合”の族——8——によって、 $M$ のすべてのコンパクト集合を含む $M$ の部分集合の最小の $\sigma$ 有限・ $\sigma$ 加法族のメンバーが意味付けられる。 $\mu(B)$ は集合 $M$ のBorel 部分集合の族 $B$ 上で定義された $\sigma$ 有限・ $\sigma$ 加法・実数値測度であるとする。特徴画像関数 $\eta(x_\mu, y_\mu, t_\mu)$ ,  $(x_\mu, y_\mu, t_\mu) \in M$ は、 $\mu$ 可測であるとする、それによってこの特徴画像関数は、局所的凸型線形位相空間 $\Phi(R^3)$ 上で定義される連続線形汎関数 $\Gamma_\eta$ （一般関数）を形成するために使うことが出来る。検定関数 $\phi \in \Phi(R^3)$ における特徴画像関数 $\eta$ に関連付けられた一般関数 $\Gamma_\eta$ の値は次の関係：——9——によって与えられる。異なる特徴画像関数 $\eta(x_\mu, y_\mu, t_\mu)$ に関連付けられた一般関数 $\Gamma_\eta$ は、族——10——に一体化される。ここで表記を簡単にするために、記号 $\eta$ は特徴画像関数のための表記法の一部としてのその役割に加えてある特定の特徴画像関数 $\eta(x_\mu, y_\mu, t_\mu)$ に関連付けられた一般関数 $\Gamma_\eta$ を指定するインデックス（助変数）の役割をする。記号 $H$ はインデックスの集合を表す。

【0038】局所的凸型線形位相空間 $\Phi(R^3)$ 上で定義される一般関数 $F$ が与えられ、負でない整数の定数 $m_r, m_y, m_t$ が与えられると、一般関数 $F$ の“一般偏導関数”——11——は、以下のとおり局所的凸型線形位相空間 $\Phi(R^3)$ 上で定義される一般関数——12——となる。検定関数 $\phi \in \Phi(R^3)$ における一般関数——12——の値は、検定関数——13——における一般関数 $F$ 自身の値となる。

【0039】“組合せ一般初期画像関数”は、輝度画像関数と同伴の一般関数の一般偏導関数と特徴画像関数と同伴の一般関数の一般偏導関数との線形結合として定義される。——14——は実数値定数の集合であるとし、 $g(\lambda)$ は集合 $\lambda$ に付随したインデックスであるとする。そして——15——は定数の集合 $\lambda$ に対応する負でない整数定数の集合であるとする。それから、この定数集合 $\lambda$ に対応する組み合わせ一般初期画像関数は、局所的凸型線形位相空間 $\Phi(R^3)$ 上で定義された一般関数 $\Gamma_{g(\lambda)}$ である。検定関数 $\phi \in \Phi(R^3)$ における定数集合 $\lambda$ に対応する組み合わせ一般初期画像関数 $\Gamma_{g(\lambda)}$ は、——16——という関係によって与えられ、そ

して検定関数 $\phi$ 上の定数集合 $\lambda$ に対応する組み合わせ一般初期画像関数 $\Gamma_{g(\lambda)}$ の“観測”と呼ばれる。異なる値を持つ定数集合 $\lambda$ に対応する組み合わせ一般初期画像関数 $\Gamma_{g(\lambda)}$ は、——17——という族に一体化される。

【0040】ここで記号 $\lambda$ は異なる定数集合 $\lambda$ すべての族を表す。一方、記号 $g$ は、族 $\Lambda$ から記号 $G$ によって表されるインデックスの集合への一対一の写像を表す。以下では表記を簡単にするために、組み合わせ一般初期画像関数のための表記に現れる引き数は省略され、それを表すためには、記号 $\Gamma_{g(\lambda)}$ ,  $\lambda \in \Lambda$ の代わりに記号 $\Gamma_g$ ,  $g \in G$ が使われる。

【0041】“測定関数”と呼ばれる所定の固定検定関数——18——が存在すると仮定する。（測定関数 $\Psi$ の例は後述する。） $g \in G$ とする。それから、ある凸型有界部分集合——19——のすべての点 $(x, y)$ （こういう点は“画像点”と呼ばれる）とすべての視点 $t \in T$ と助変数——20——のすべての値とに関して、画像点 $(x, y)$ における助変数値 $\sigma$ と視点 $t$ に対応する画像の成分——21——の値は、検定関数——22——、——23——上の組み合わせ一般初期画像関数 $\Gamma_g$ の観測——24——として決定される。表記を簡単にするために記号 $g$ は、画像の成分のためのインデックスとしての役割に加えて、助変数値 $\sigma$ に対応する画像の成分に関する表記——25——の役に使われるということに注意すること。

【0042】集合——26——からのすべての助変数値 $\sigma$ に関して関係（2-4）によって指定される検定関数——27——上の組み合わせ一般初期画像関数 $\Gamma_g$ の観測——28——として集合 $\Omega \times T$ 上で定義された各成分——29——を持った助変数ベクトル値関数——30——（ここで記号 $\Omega \times T$ は集合 $T$ 上の集合 $\Omega$ のカルデシアン積を表す）は、“助変数視点変化画像関数”と呼ばれる。集合 $\Omega \times T$ からの点 $(x, y, t)$ は、“視点変化画像点”と呼ばれる。

【0043】助変数値——31——に対応し、かつ視点 $t \in T$ から取られた“画像”は、助変数視点変化画像関数——32——の値の集まり——33——を意味する。すべての助変数値——34——に対応し、かつ固定視点 $t \in T$ から取られた画像——35——の集まり——36——は、視点 $t$ から取られた“助変数画像”と呼ばれる。

【0044】助変数視点変化画像関数——37——の各成分——38——は、定義域 $\Omega \times T$ 内のいたるところで無限に微分可能であって、変数 $x, y, t$ に関するその偏導関数は、関係（2-4）によって指定される検定関数——39——の助変数 $x, y, t$ に関する偏導関数上の組み合わせ一般初期画像関数 $\Gamma_g$ の観測として得られる。例えば、助変数視点変化画像関数——40——の第1階偏導関数——41——の成分——42——は、すべての——43——に関して、検定関数——44——上の組み合わせ一般初期画像関数 $\Gamma_g$ の観測——45——として与えられる。こ

---47---であって、 $\Psi_1(x, y, t)$ ,  $\Psi_2(x, y, t)$  は変数  $x, y$  に関する関数  $\Psi(x, y, t)$  の偏導関数である。

【0045】---48---を連続した3個の視点(ここで\*

$$U_A(x, y, t, \Delta t^-, \Delta t^+) \\ = (x(t + \Delta t^+) - x(t - \Delta t^-)) / (\Delta t^- + \Delta t^+) \quad (2-7)$$

$$V_A(x, y, t, \Delta t^-, \Delta t^+) \\ = (y(t + \Delta t^+) - y(t - \Delta t^-)) / (\Delta t^- + \Delta t^+) \quad (2-8)$$

但し、

$$x = (\Delta t^- x(t + \Delta t^+) + \Delta t^+ x(t - \Delta t^-)) / (\Delta t^- + \Delta t^+) \quad (2-9)$$

$$y = (\Delta t^- y(t + \Delta t^+) + \Delta t^+ y(t - \Delta t^-)) / (\Delta t^- + \Delta t^+) \quad (2-10)$$

によって定義されるベクトル  $\{U_A(x, y, t, \Delta t^-, \Delta t^+), V_A(x, y, t, \Delta t^-, \Delta t^+)\}$  は、視点  $t + \Delta t^+$  と視点  $t - \Delta t^-$  からのシーン投影に基づく点---51---と視点  $t \in R$  におけるシーン投影の“平均速度ベクトル”と呼ばれる。ベクトル  $\{U_A(x, y, t, \Delta t^-, \Delta t^+), V_A(x, y, t, \Delta t^-, \Delta t^+)\}$  は、視点  $t + \Delta t^+$  のみならず視点  $t - \Delta t^-$  から見えるシーン内の点の投影である集合  $R^2$  からのこれらの点  $(x, y)$  においてのみ定義できる。こういう画像点の集まりは記号  $\Omega(t, \Delta t^-, \Delta t^+)$  によって表される。

【0046】さて、点---52---と点---53---におけるシーン投影の“瞬間速度ベクトル”  $\{u_1(x, y, t), v_1(x, y, t)\}$  を定義しよう。 $\Delta t^-, \Delta t^+$  は正の視点増分であるとし、また  $W(x, y, t, \Delta t^-, \Delta t^+)$  は---54---の場合に平均速度ベクトル  $\{U_A(x, y, t, \Delta t^-, \Delta t^+), V_A(x, y, t, \Delta t^-, \Delta t^+)\}$  を含む単一集合に等しく、そうでない場合には空集合  $\emptyset$  に等しい集合であるとする。それからすべての正の視点増分  $\Delta t < \delta t$  (ここで  $\delta t$  は正の実数定数) に関して、集合  $W(x, y, t, \Delta t)$  は関係---55---から定義され一方、集合  $W(x, y, t)$  は関係---56---から定義される。ここで記号---57---は集合  $W(x, y, t, \Delta t)$  の位相閉包を意味する。 $W(x, y, t)$  は殆どすべての  $(x, y, t) \in R^3$  に関して単一集合であると仮定している。この仮定は、集合  $W(x, y, t)$  が単一集合でない場合、点  $(x, y, t)$  を含む集合  $R^3$  の部分集合はゼロに等しいLebesgue測度を持つということの意味している。集合  $W(x, y, t)$  が単一集合からなっているようなすべての点  $(x, y, t) \in R^3$  に関して、その元は瞬間速度ベクトル  $\{u_1(x, y, t), v_1(x, y, t)\}$  の値として選択され、そうでない場合はゼロベクトルが瞬間速度ベクトル  $\{u_1(x, y, t), v_1(x, y, t)\}$  の値として選択され

\*  $\Delta t^-$  と  $\Delta t^+$  は正の視点増分である)とし、また---49---を視点  $t - \Delta t^-$  から取られたシーン内のある点の投影であるとし一方、---50---は視点  $t - \Delta t^+$  から取られた同じ点の投影である。関係

る。

【0047】関数---58---は  $R^3$  内のLebesgue測度  $dx dy dt$  に関して局所的に積分可能であり、またそれ故に、局所的凸型線形位相空間  $\Phi(R^3)$  上で定義された連続線形汎関数  $U_1, V_1$  (一般関数)を形成するために使うことができる。検定関数  $\phi \in \Phi(R^3)$  における一般関数  $U_1, V_1$  の値は次の関係:---59---

---60---によって与えられる。 $(x, y)$  を  $\Omega$  からの画像点、 $t$  を  $T$  からの視点、 $\sigma$  を---61---からの助変数値とすると、点  $(x, y)$  と視点  $t$  における助変数値  $\sigma$  に対応する画像の“速度ベクトル”---62---は次の関係---63---

---64---定義される。ここで  $\Psi$  は集合  $\Phi(R^3)$  からの測定関数である。前に述べたように速度ベクトル---65---の垂直成分---66---は画像点  $(x, y)$  から独立であって、それ故に---67---と表され一方、速度ベクトルは---68---と表される。視点  $t \in T$  と助変数値  $\sigma \in [1, \infty)$  とに対応する速度ベクトル---69---の集まりは、助変数値  $\sigma$  に対応しかつ視点  $t$  において取られた画像---70---の“速度ベクトル場”と呼ばれる。すべての助変数値  $\sigma \in [1, \infty)$  に対応しかつ固定視点  $t \in T$  において取られた画像---71---の速度ベクトル場---72---の集まり---73---は、視点  $t$  において取られた助変数画像---74---の“助変数速度ベクトル場”と呼ばれる。---75---をある視点間隔  $T$  からの有限増加視点列であるとし、また  $t$  を同じ視点間隔  $T$  からの一つの視点であるとする。

【0048】そうすると評価問題は下記のように定式化できる。“助変数視点変化画像列”と呼ばれる、視点---76---において取られた助変数画像の列---77---と視点  $t$  とを与えて、視点  $t$  において取られた助変数画像---78---の助変数速度ベクトル場---79---の評価である助変数ベクトル場---80---を求める。

【0049】助変数速度ベクトル場---81---への立ち上がりを与えるシーン内の変化は助変数視点変化画像列

---82---に反映されるけれども、それらの間の関係は必ずしも一意ではない。同じ助変数視点変化画像列が異なる助変数速度ベクトル場に関連付けできることがあり、またその反対に同じ助変数速度ベクトル場が異なる助変数視点変化画像列から生ずることもある。

【0050】それから本発明の方法は、所定の助変数視点変化画像列に対応する助変数速度ベクトル場の評価を決定することを含んでいる。あり得べきこの決定のために、撮像されるシーンとその撮像プロセスそのものについて幾つかの特別の仮定を置かなければならない。筆者らが置いた仮定を次に述べる。これらの仮定に基づいて、助変数速度ベクトル場の評価には幾つかの制約が課せられる。そうするとその評価の決定は、所定の助変数視点変化画像列に関するこれらの制約から得られる連立方程式を解くことに帰着する。

【0051】---83---を、視点間隔 $T$ 内にあって増加していく列を形成する視点---84---から撮影された所定の助変数視点変化画像列であるとし、また $t$ を視点間隔 $T$ 内のある所定の視点であるとする。助変数速度ベクトル場---85---の評価---86---を助変数視点変化画像列---87---の関数として求める問題を考察する。

【0052】下記において、助変数速度ベクトル場---88---の評価---89---は、一組の制約を満足する助変数速度ベクトル場として決定される。これらの制約は、以下の仮定に基づいている。

【0053】1. 撮像されるシーンは、略一定で略様な照明を受けているものとする。これは、シーン内の各表面片の投射照明の変化はその空間全体にわたって小さいということと、その変化は主として光源に対するこれら表面片の向きによるものであるということとを意味する。

【0054】2. シーン内の各表面点の輝度は略Lambertianであると仮定する。これは、局所的には観察位置への依存度が小さく、無視できるということを意味している。

【0055】3. 前節で述べたように、ある視点 $t \in R$ における画像平面 $R^2$ 内の各点 $(x, y)$ の輝度 $\xi(x, y, t)$ 、 $\xi \in \Xi$ は、視点 $t$ における画像平面内の点 $(x, y)$ に投影しているシーン内の点の輝度に略比例し、その比例定数は位置 $(x, y)$ と視点 $t$ とは無関係であるということを仮定している。

【0056】4. シーンは不透明な物体からできていると仮定している。特徴点---91---を選択するための基準と特徴値---90---を各特徴点---91---に割り当てるための基準は、これらの物体の真の性質を表しており、物体の空間的姿勢には無関係であるということを仮定している。

【0057】5. シーン内の物体の近傍点の速度は、一方では類似しており、他方では視点に関してゆっくりと変化すると仮定している。言い換えれば、シーン内の各

物体の表面点の助変数視点変化速度ベクトル場は、その空間全域そして視点位置全域にわたって滑らかに変化すると仮定している。

【0058】 $\Pi_i$ は関係---92---によって定義される空間 $R^2$ の部分集合、 $P_i$ は集合 $\Pi_i$ の部分集合、---93---は集合 $P_i$ 上の集合 $G$ のCartesian積であるとする。集合 $G_i$ からの元は記号 $g_i$ によって表され、また集合 $G_i$ は集合 $G_i$ のBorel部分集合の族 $B(G_i)$ 上で定義されたある測度 $d g_i$ を有する可測空間であると仮定している。実際問題として、集合 $G_i$ は有限であって測度 $d g_i$ は有限集合 $G_i$ 上のある点測度であるが、表現の一様性のためにさらに一般的な手法が使われる。

【0059】すべての $(x, y) \in \Omega$ に関して、---94---であるとし、また---95---は関係---96---によって定義される関数であるとする。ここで---97---は、画像点 $(x, y)$ と視点 $t$ における助変数値 $\sigma$ に対応する画像の速度ベクトルである。ここで表記を簡単にするために、記号 $g_i$ は前に定義された関数(3-2)のためのインデックスとしての役割に加えてこういった関数のための表記---98---の役割に用いられる。

【0060】 $(x, y) \in \Omega$ は、遮蔽境界に属さないシーン内のある点の視点 $t$ において取られた投影である画像点であるとする。ある物体の遮蔽境界は、物体のシルエットに投影する物体のその部分に属するシーン内の点と定義される。本節の初めになされた仮定1~4は、関数---99---の絶対値が小さいということを暗示している。したがって、その最小値が助変数値 $\sigma$ に対応する画像の速度ベクトル---100---の評価---101---を指定する汎関数の一部として、関数---102---を使うことは当然である。関数(3-3)は習慣的に、助変数値 $\sigma$ とインデックス $g_i$ 、[2, 12~14, 18~21]とに対応する“光学流れ制約”と呼ばれる。

【0061】もし視点増分 $\Delta t^-$ 、 $\Delta t^+$ がゼロに値が近づくならば、2に等しいか2より大きい整数 $n$ に関して速度ベクトルの評価の成分---103---に関する関数---104---の第 $n$ 階偏導関数は---105---に比例する割合でゼロに近づくことに注意すること。これは特に、視点増分 $\Delta t^-$ 、 $\Delta t^+$ がゼロに近づくとき関数---106---は速度ベクトルの評価の成分---107------に関してリニアな関数に近づく。

【0062】関数(3-3)がその最小値に達する速度ベクトルの評価---108---が連立方程式---109---を満足するということは容易に検証できる。ここで関数---110---と---111---とは速度ベクトルの評価の成分---112---と---113---とに関する関数---114---の第1階偏導関数である。

【0063】ベクトル---115---を関係---116------117---のように定義する。速度ベクトル---118------について関係(3-4)の左辺をテイラー級数に展開し、そして速度ベクトルの成分---119---と---120---

— とに関する関数——121—— の第2階以上の高階偏導関数をゼロにセットすることにより、ベクトル——122———に関する連立線形方程式——123———を得る。ここで関数——124———と——125———は、速度ベクトルの成分——126———と——127———とに関する関数——128———の第1階偏導関数である。

【0064】初めに速度ベクトルの成分——129———と——130———とに関して関数——131———の勾配——132———のノルムは相対的に大きいと仮定し、それから連立方程式(3-7)が関係——133———と同等であると仮定しよう。この最後の関係は、関数(3-3)が速度ベクトル——134———の評価——135———についてある最小値に達するための要件が勾配(3-8)の方向の評価の成分を指定し、勾配(3-8)に直交する成分を未決定のままにしておくということを暗示している。

【0065】もし勾配(3-8)のノルムが比較的小さければ、連立方程式(3-7)はあまりに弱くなり過ぎ、また関数(3-3)が速度ベクトル——136———の評価——137———についてある最小値に達するための要件はその評価についてなんの制約も課さない。

【0066】上記の論議は、計算をうまく定義するためには助変数速度ベクトル場——138———の評価——139———について追加の制約を課する必要があるということを示唆している。

【0067】——140———は画像平面内で方向を指定する各ベクトル——141———を有する、二次元ユークリッド空間 $R^2$ 内の単位円であるとしよう。集合 $S$ は、集合 $S$ のBorel部分集合の族 $B(S)$ 上で定義されたある一様な測度 $ds$ を有する可測空間であると仮定する。

【0068】すべての $(x, y) \in \Omega$ に関して、——142———であるとし、また——143———は関係——144———によって定義される関数であるとする。ここで記号 $\nabla$ は変数 $x, y$ に関するある関数の勾配を表す。

【0069】——145———であるとし、また $(x, y) \in \Omega$ は、遮蔽境界を $s$ 方向に横切らないシーン内のある点の視点 $t$ において取られた投影である画像点であるとする。これは次のことを意味している。すなわちすべての十分に小さい正の定数 $\omega$ について、シーン内の次の2個の点は両者とも同一の物体に属して互いに十分に近接しているものとし、第一の点は視点 $t$ において画像点——146———に投影しているものであり、これに対して第二の点は視点 $t$ において画像点 $(x, y)$ に投影しているものであるとする。本節の初めになされた仮定5は、関数——147———の絶対値が小さいということを暗示している。したがって、その最小値が助変数値 $\sigma$ に対応する画像の速度ベクトル——148———の評価——149———を指定する汎関数の一部として、関数——150———を使うことは当然である。関数(3-11)は習慣的に、方向 $s$ [2, 12~14, 18~21]の“平滑制約”と呼ばれる。

【0070】最後に、助変数速度ベクトル場——151———の評価——152———を計算するプロセスをうまく定義するためには、光学流れ制約と方向平滑制約とがなんら制限を課さないようなすべての画像点 $(x, y) \in \Omega$ とすべての助変数値 $\sigma \in [1, \infty)$ とに関して速度ベクトル——153———の未知の評価——154———とその初期評価——155———との間の差異が小さいということを必要とする。これは、関数(3-12)と(3-13)とに結び付けられた重さは関数(3-3)と(3-11)とに結び付けられた重さに比較して小さいと仮定した上で、光学流れ制約と方向平滑制約とを指定する関数(3-3)と(3-11)とに加えてその最小値が助変数値 $\sigma$ に対応する画像の速度ベクトル——156———の評価——157———を指定する汎関数の中に、関数——158———

——159———を含めることによって達成できる。関係(3-12)と(3-13)の中に現れる初期評価——160———は本節の中で後述する。上記の要件によって画像点 $(x, y) \in \Omega$ と視点 $t$ における速度ベクトル——161———の評価——162———に課せられた制約は“正規化制約”と呼ばれる。

【0071】先に述べたように、本節の初めに述べた仮定が守られたときでも、光学流れ制約と方向平滑制約は遮蔽境界の近傍の点で必ずしも有効でない。本発明の助変数速度ベクトル場の評価を計算する方法は、制約が有効でないときにはいつでも各制約と結び付けられた重さが小さくなるような仕方でも各制約と結び付けられた重さを調節することにより、上記の困難を解決する。光学流れ制約、方向平滑制約、および正規化制約を指定する関数(3-3)、(3-11)、(3-12)、および(3-13)はどのようにして助変数速度ベクトル場の評価の汎関数に結合されるかを以下に説明する。それからこの評価は、このような汎関数に関連したある一定の最適性基準から得られる連立非線形方程式を解くことによって計算される。

【0072】次に、速度ベクトル場の評価において本発明の方法で用いられる連立非線形方程式を説明する。本発明の方法では、助変数速度ベクトル場——163———の評価——164———は、光学流れ制約、方向平滑制約、および正規化制約の重み付き平均が最小になる助変数速度ベクトル場として決定される。重み関数は、遮蔽境界の近傍にあるすべての画像点 $(x, y) \in \Omega$ において次の要件が満足されるような仕方では選ばれる。すなわち光学流れ制約に結び付けられた各重み関数はその光学流れ制約が満足されないときはいつでも小さくなり、また遮蔽境界を横切る方向に対応する平滑制約に結び付けられた各重み関数は方向平滑制約が満足されないときはいつでも小さくなる。ある所定の画像点の近傍の遮蔽境界の存在のための基準と、ある所定の画像点の近傍で所定の方向に横切られる遮蔽境界のための基準とは、助変数速度ベクトル場の未知の評価の値によって最も有効に定義で

きる。したがって、助変数速度ベクトル場の未知の評価の値は、暗黙に各重み関数に使わなくてはならない。一方、各重み関数は、対応する制約の相対的重要性を重み付き平均の一部として指定するだけであって制約それ自身を指定しないから、あたかも助変数速度ベクトル場の未知の評価の値とは無関係であるかのように取り扱わなくてはならない。これらの困難を克服するために、助変数速度ベクトル場の未知の評価の二つのコピー、不変コピーと可変コピーとが導入される。重み関数では助変数速度ベクトル場の未知の評価の不変コピーの値が使われ、これに対して制約関数では助変数速度ベクトル場の未知の評価の可変コピーの値が使われる。それから光学流れ制約、方向平滑制約、および正規化制約の重み付き平均に対してエネルギー最小化原理に反するようなさらに一般的な変分原理を適用して、助変数速度ベクトル場の未知の評価の連立非線形方程式を導き出す。この変分原理は次のように説明できる。

【0073】—165— は助変数速度ベクトル場の未知の評価の不変コピーであり、—166— は助変数速度ベクトル場の未知の評価の可変コピーであり、そして助変数ベクトル場—167— は関係—168—によって定義されるものであるとする。

【0074】画像 $\Omega$ の境界—169—に直角な方向の助変数ベクトル場—170—の導関数は、すべての助変数値 $\sigma \in [1, \infty)$ に関して、また画像境界—171—に属するすべての画像点 $(x, y)$ に関してゼロに等しい。このことは、助変数速度ベクトル場の未知の評価の可変コピーに適用できる変分の型に制限を課することになる。

【0075】—172—はそれぞれ次の関係—173—によって光学流れ制約、方向平滑制約、および正規化制約を指定する関数(3-3)、(3-11)、(3-12)、および(3-13)の重み付き平均として定義される助変数ベクトル場—174—の汎関数であるとする。ここで、

(i) —175—は、助変数値 $\sigma$ とインデックス $g$ とに対応する光学流れ制約に結び付けられた重みであり、これは独立変数 $x, y, t$ の関数であり、またこれらの変数の関数—176—の関数である。

【0076】(ii) —177—は、 $s$ 方向の平滑制約に結び付けられた重みであり、これは独立変数 $x, y, t$ の関数であり、またこれらの変数の関数—178—の関数である。

【0077】(iii)  $\gamma_{\gamma, s}$ は、正規化制約のための重みを指定する正の定数である。

【0078】(iv) 関数—179—は、ベクトル値関数—180—のノルムであって、これは関係—181—によって与えられる関数—182—の変数 $x, y$ に関する勾配である。

【0079】(v) 関数—183—は、関係(3-10)

のように定義される。

【0080】(vi) 関数—184—は、関係(3-2)のように定義される。そして

(vii) 画像点 $(x, y) \in \Omega$ における助変数値 $\sigma \in [1, \infty)$ に対応する速度ベクトルの初期評価の成分—185—は後で定義する。

【0081】それから助変数速度ベクトル場の評価は、助変数ベクトル場—186—の値が助変数ベクトル場—187—の値とまったく等しいときに、助変数ベクトル場—188—の関数と見なされかつ助変数に依存すると同様に助変数ベクトル場—189—に依存している汎関数—190—が局所的最小値に達する助変数ベクトル場—191—として定義される。関係(4-1)を考慮に入れると、汎関数—192—は—193—の形で表すことができる。助変数ベクトル場—194—は助変数ベクトル場—195—に対して摂動を指定し、また汎関数—196—は助変数ベクトル場—197—とその摂動—198—の各選択に対してコストを割り当てる。それから助変数速度ベクトル場の評価は、助変数ベクトル場—199—となる。これに関しては、摂動—200—がまったくゼロに等しいとき局所的最小コストが得られる。汎関数—201—に適用される助変数ベクトル場—202—に関して変分法を使うと、助変数速度ベクトル場の前記で定義された評価—203—は、連立方程式—204——205—の解となる。ここで関数—220—と—221—は、速度ベクトルの評価の成分—223—と—224—に関する関数—222—の第1階偏導関数である。

【0082】光学流れ制約、方向平滑制約、および正規化制約に結び付けられた重み関数を選択するためには、助変数ベクトル場—226—が助変数ベクトル場—227—に完全に等しく、またこれら両者が連立非線形方程式(4-5)の解として得られた助変数速度ベクトル場の評価—228—に等しいという条件の下で関係(4-2)によって定義される汎関数—225—を考慮する。この場合、汎関数—229—は—230—の形を取る。—231—は連立非線形方程式(4-4)の解として得られた助変数速度ベクトル場の評価であるとする。光学流れ制約—234—が満足されないような遮蔽境界の近傍に位置するすべての画像点 $(x, y) \in \Omega$ に関して汎関数(4-5)に対する光学流れ制約—233—の寄与が小さくなるような仕方でも重み関数—232—の各々を選択する。同様に、重み関数—235—の各々は、遮蔽境界が $s$ 方向に横切られ、方向平滑制約—237—が満足されないような遮蔽境界の近傍に位置するすべての画像点 $(x, y) \in \Omega$ に関して汎関数(4-5)に対する方向平滑制約—236—の寄与が小さくなるような仕方でも選択される。画像点 $(x, y) \in \Omega$ を遮蔽境界の近傍に置くと、次の二つの事象が発生し

易くなる。

【0083】1. 幾つかの正の視点増分 $\Delta t^-$ と $\Delta t^+$ とに関して、視点 $t$ において画像点 $(x, y)$ に投影しているシーン内の点は、視点 $(t - \Delta t^-)$ からは見えて視点 $(t + \Delta t^+)$ からは見えないか、視点 $(t + \Delta t^+)$ からは見えて視点 $(t - \Delta t^-)$ からは見えないかのいずれかである。 $(x_1, y_1)$ と $(x_2, y_2)$ はそれぞれ、視点 $(t - \Delta t^-)$ と $(t + \Delta t^+)$ から取られたシーン内のこういう点の画像平面への投影であるとする。もし視点 $(t - \Delta t^-)$ において画像点 $(x_1, y_1)$ に投影しているシーン内の点の輝度が視点 $(t + \Delta t^+)$ において画像点 $(x_2, y_2)$ に投影しているシーン内の点の輝度と大きく異なっているならば、またもしその輝度がシーン内の上記の点の近傍で変化するならば、下記の場合が起こりそうである。

【0084】1.1. 視点 $(t - \Delta t^-)$ において画像点——238——に投影しているシーン内の点と視点 $(t + \Delta t^+)$ において画像点——239——に投影しているシーン内の点の両者は、一方では遮蔽された物体に属しており、他方では異なる輝度を持っている。後のことは、上記の点がそのシーンの物体の異なる位置を占めている場合に起こるかもしれないが、これはまた遮蔽境界の存在によっても起こりうる。これらの条件の下で関数——240——の中の幾つかの関数の絶対値が大きくなる。

【0085】1.2. 視点 $(t - \Delta t^-)$ において画像点——241——に投影しているシーン内の点と視点 $(t + \Delta t^+)$ において画像点——242——に投影しているシーン内の点の両者は、遮蔽している物体に属しているか、遮蔽されている物体に属しているかのいずれかである。

【0086】これに加えて、あるあまり大きくすぎないベクトル——243——に関して、シーン内の下記の二つの点の一つは遮蔽している物体に属しており、他の点は遮蔽されている物体に属している。すなわち第一の点は視点 $(t - \Delta t^-)$ において画像点——244——に投影している点であり、第二の点は視点 $(t + \Delta t^+)$ において画像点——245——に投影している点である。同様にして、同じベクトル——246——に関して、シーン内の下記の二つの点の一つは遮蔽している物体に属しており、他の点は遮蔽されている物体に属している。すなわち第一の点は視点 $(t - \Delta t^-)$ において画像点——247——に投影している点であり、第二の点は視点 $(t + \Delta t^+)$ において画像点——248——に投影している点である(図42を参照)。こういう場合、関数——249——の値は比較的小さくなるが、関数——251——の中の幾つかの関数の値ばかりでなく関数——250——の中の幾つかの関数の値も比較的大きくなる。これは、関数——252—— $g_i \in G_i$ の中の幾つかの関数の値が大きくなるということを暗示している。もし速度ベクトルの評価の成分——254——と——255——とに関する関数

——253——の第二階以上の高階偏導関数が無視されるならば、関数(4-7)の各々は——256——という形で表すことができる。ここで関数——257——と——258——は速度ベクトルの評価の成分——260——と——261——とに関する関数——259——の第一階偏導関数である。ベクトル——262——と共線となっているベクトル $s \equiv (s_x, s_y)$ 、 $s \in S$ に関して上記の観測は、関係——264——によって定義される関数——263——の中の幾つかの関数の絶対値が大きいうことを暗示している。

【0087】2. 助変数速度ベクトル場の評価——265——は、画像点 $(x, y)$ の近傍で急速に変化する(図2を参照)。これは次のことを意味する。すなわち、

2.1. 関数——266——の値が大きくなる。

【0088】2.2. もし画像点 $(x, y)$ が $s$ 方向に遮蔽境界を横切るならば、関数——267——の絶対値が大きくなる。

【0089】もし撮像されているシーンと撮像プロセスそのものに関する、前節の初めに述べた条件が満足されるならば、ある $g_i \in G_i$ に関して関数——270——の絶対値が大きいうに加えて、ある $g_i \in G_i$ に関して関数——268——の絶対値が大きいかあるいは関数——269——の絶対値が大きいう場合、画像点 $(x, y)$ は遮蔽境界の近傍にある可能性が高く、また一方では関数——273——の絶対値が大きいうに加えて、ある $g_i \in G_i$ に関して関数——271——の絶対値が大きいかあるいは関数——272——の中の幾つかの関数の絶対値が大きいう場合、画像点 $(x, y)$ は $s$ 方向に遮蔽境界を横切る可能性が高い。関数——274——が大きいう場合、画像点 $(x, y)$ は必ずしも遮蔽境界の近傍にはないということに注意すること。例えば、その画像点は、画像平面への投影が回転または局所的変形受けるようなシーン内の物体上にあるかもしれない。また関数——275——の中の幾つかの関数が大きいう場合、画像点 $(x, y)$ は必ずしも $s$ 方向に遮蔽境界を横切らないということにも注意すること。例えば画像点は、テクスチャーまたは照明の突然の変化から生ずる輝度境界を横切る可能性もある。

【0090】これらの観測は以下のことを示唆している。すなわち、重み関数——276——の各々は、関数——277——の絶対値に比較してまた関数——279——の絶対値を乗じた関数——278——の値に比較して定常的に減少する関数であるべきであり、また重み関数——280——の各々は、関数——281——の絶対値に比較してまた関数——283——の絶対値を乗じた関数——282——の中の幾つかの関数の絶対値に比較して定常的に減少する関数であるべきである。

【0091】制約関数が大きくなるときはいつでも、重み関数と汎関数(4-6)内に加数として現れるその対応する制約関数との積も大きくなるような仕方では選ばれる

光学流れ制約か方向平滑制約のいずれかに対応する重み関数の各々を必要とすることは当然である。これは、助変数速度ベクトル場の初期評価の選択に関して、また雑音に関してさらに強い助変数速度ベクトル場の評価を決定する連立非線形方程式(4-4)の解を与える。さらに詳しくは、これらの要件は下記のことからなる。

【0092】1. すべての  $g_i \in G_i$  に関して、関係—284—によって定義されかつ汎関数(4-5)内に加数として現れる関数は、関数—285—と比較して定常的に増加していくことが必要である。言い換えれば、関数(4-9)は、関数—286—が増加するときはいつでも増加する必要がある。

【0093】2. すべての  $s \in S$  に関して、関係—287—によって定義されかつ汎関数(4-5)内に加数として現れる関数は、関数—288—と比較して定常的に増加していくことが必要である。言い換えれば、関数(4-10)は、関数—289—が増加するときはいつでも増加する必要がある。

【0094】 $r^2$ 、 $p^2$ 、 $q^2$  が負でない実数定数の場合、すべての  $g_i \in G_i$  に関して、重み関数—290—は、また式—292—が関係—293—のように定義されかつ—294—が負でない実数定数の場合、すべての  $s \in S$  に関して、重み関数—291—は、上記の要件および本節の初めのほうで述べた要件に従う。

【0095】各助変数値  $\sigma$  に関して独立に連立非線形方程式(4-4)を解くことによって助変数速度ベクトル場—295—の評価得るという手法は、一般に連立方程式(4-4)が多くの解を持っている可能性があるということに関連する困難に遭遇する可能性がある。この困難を克服するために、助変数  $\sigma$  の異なる値に対応する速度ベクトル場の評価の値は、下記の制限を課することによって一緒に結合される。すなわち、助変数速度ベクトル場—296—の評価は、すべての画像点  $(x, y) \in \Omega$  とすべての助変数値  $\sigma \in [1, \infty)$  とに関して連続的に微分可能な関数であることが必要である。助変数速度ベクトル場—297—の評価についての追加の制限は、下記の境界条件の形で課せられる。すなわち、

1. 助変数速度ベクトル場—298—の評価は、助変数  $\sigma$  が  $\infty$  に収束するとき所定のベクトル定数—299—に完全に等しいベクトル場に収束する必要がある、またこういった収束は画像点の集合  $\Omega$  全域にわたって一樣であることが必要である。

【0096】2. 画像  $\Omega$  の境界  $\partial \Omega$  に直角な方向の助変数速度ベクトル場—300—の評価の偏導関数は、画像境界  $\partial \Omega$  に属するすべての画像点  $(x, y)$  とすべての助変数値  $\sigma \in [1, \infty)$  とに関してゼロに等しいことが必要である。

【0097】 $\sigma \in [1, \infty)$  は所定の助変数値であるとし、 $\Delta \sigma$  は無限小の正の実数値であるとする。上記の制限を考慮に入れると、助変数値  $\sigma$  に対応する速度ベクトル

ル場—301—の初期評価は、次のように定義できる。すなわち、—302—それから正規化制約(3-12)と(3-13)はそれぞれ、すべての画像点  $(x, y) \in \Omega$  に関して関係—303—

—304—によって近似できる。ここで  $r$ 、 $p$ 、 $q$  は関係—305—

—306—によって与えられる正の実数である。

【0098】連立非線形方程式(4-4)に現れる正規化制約—307—と—308—とをそれぞれ制約—309—と—310—とで置き換えることにより、また関係(4-11)と(4-12)とにしたがって重み関数—311—と—312—とを定義することにより、連立方程式(4-4)は助変数速度ベクトル場—313—の未知の評価に関して下記の連立方程式—314—になる。

【0099】関係(4-2)によって定義される汎関数—316—に現れる正規化制約—315—と—315-1—とをそれぞれ制約—317—と—318—とで置き換えるという条件の下で、そして重み関数—319—は関係(4-11)にしたがって定義されるが、これに対して重み関数—320—は関係(4-12)にしたがって定義されるという条件の下で、関係(4-2)によって定義される助変数ベクトル場—322—と助変数ベクトル場—323—との汎関数—321—は、関係—326—のように定義される同じ助変数速度ベクトル場—325—の汎関数—324—で置き換え可能である。ここで記号  $\ln$  は自然対数関数を表す。

【0100】もし助変数ベクトル場—330—が助変数ベクトル場—331—にまったく等しいときにはいつでも助変数ベクトル場—329—に関する汎関数—328—のすべての変分がゼロに等しければ、そしてただそうでさえあれば、助変数ベクトル場—327—は連立非線形方程式(4-19)を満足するということを、上記の代替は暗示している。

【0101】定数  $q^2$  がゼロに近づくとき、そして視点列—332—の逐次視点間の増分が一樣にゼロに近づくとき、汎関数—333—は、助変数ベクトル場—334—とは独立の汎関数に近づき、そして助変数速度ベクトル場の評価を求める問題はエネルギー最小化問題になる。

【0102】—335—は連立非線形方程式(4-19)の解として得られる助変数速度ベクトル場の評価であるとする。こういった解の近傍での連立非線形方程式(4-19)の振る舞いを吟味する。まず初めに、画像点  $(x, y)$  が遮蔽境界を越えた位置から遮蔽境界の近傍の位置へ行くときの光学流れ制約と方向平滑制約の振る舞いの変化を考察する。それから関係(4-11)のように重み関数—336—を指定する助変数  $r^2$ 、 $p^2$ 、 $q^2$  と関係(4-12)のように重み関数—338—を指定する助変数—337—とへの光学流れ制約と方向平滑制約の

依存性を調査する。

【0103】—339—を関係(4-1)によって定義される助変数ベクトル場とすると、助変数ベクトル場—340—の近傍では、汎関数—341—は下記のように助変数ベクトル場—340—の二次形式—343—に展開できる。

【0104】インデックス  $g_i \in G_i$  に対応する光学流れ制約—344—は、式—345—によって連立非線形方程式(4-19)で表される。この式に関しては以下の場合が考慮できる。すなわち、

1. 画像点  $(x, y)$  が遮蔽境界を越えている。それから関数—346—の値は小さくなりうる。この場合、関数(4-22)の値は、関数—347—の値に近づくであろう。次にこれは、インデックス  $g_i$  に対応する光学流れ制約—348—に結び付けられた連立非線形方程式(4-19)の二つの加数の各々が、第一の加数の場合に関数—350—に等しくまた第二の加数の場合に関数—351—に等しい比例定数を持った関数—349—に関してある線形関数として振る舞うようにするであろう。

【0105】2. 画像点  $(x, y)$  が遮蔽境界に近づく。それから関数—352—の値は増加しうる。関数—352-1—の値が関数—352-2—の値より小さい場合には、光学流れ制約—353—に結び付けられた二次形式(4-21)の加数が正であるということは、関数(4-22)が増加するであろうということを暗示する。関数—354—の値が関数—355—の値より大きい場合には、光学流れ制約—356—に結び付けられた二次形式(4-21)の加数が負であるということは、関数(4-22)が減少するであろうということを暗示する。したがって、関数—358—の値が関数—359—の値に等しいときに、関数(4-22)はその最大値—357—に達する。

【0106】関数—360—の値が大きくなるということも起こりうる。これらの条件の下では下記の値、すなわち関係(4-24)によって指定される関数(4-22)の最大値とこの最大値が達成される関数—362—の値—361—とは減少する。上記の状況は下記の場合に起こりうる。

【0107】1. 遮蔽物体が、遮蔽境界に直交する方向に評価—363—を人工的に制約しうる。それから光学流れ制約は、遮蔽境界に平行な方向に速度ベクトルの評価—364—をシフトすることによって関数—365—の絶対値を減らすように反応する可能性がある。今度はこれは関数—366—の値を増加させる可能性がある。

【0108】2. 評価—367—は、光学流れ制約の非線形性の結果としてあるいは画像内に存在する雑音の結果として発散する可能性がある。

【0109】 $s \in S$  という方向の方向平滑制約—368—

—は、式—369—によって連立非線形方程式(4-19)で表される。この式に関して下記の場合が考慮できる。すなわち、

1. 画像点  $(x, y)$  は  $s$  方向に遮蔽境界を横切らない。それから関数—370—の値は小さくなりうる。この場合、関数(4-25)の値は関数—371—の値に近づくであろう。今度はこれは、方向平滑制約—372—に結び付けられた連立非線形方程式(4-19)の加数が定数値  $1/a^2$  に等しい比例定数を持った対応する制約の線形関数として振る舞うようにするであろう。

【0110】2. 画像点  $(x, y)$  は  $s$  方向から遮蔽境界に近づく。それから関数—373—の値は大きくなり得る。関数—374—の値が関数—375—の値より小さい場合には、方向平滑制約—376—に結び付けられた二次形式(4-21)の加数が正であるということは、関数(4-25)が増加するであろうということを暗示する。関数—377—の値が関数—378—の値より大きい場合には、方向平滑制約—379—に結び付けられた二次形式(4-21)の加数が負であるということは、関数(4-25)が減少するであろうということを暗示する。したがって、関数—381—の値が関数—382—の値に等しいとき、関数(4-25)はその最大値—380—に達する。

【0111】関数—383—の中の幾つかの関数の値が大きい場合、下記の値、すなわち関係(4-27)によって指定される関数(4-25)の最大値とこの最大値が達せられる関数—385—の値—384—とは減少する。こういう動きは、この場合もしこういった  $s$  方向における速度ベクトル場の評価の平滑さの度合いを指定する関数—386—の値が大きければ画像点  $(x, y)$  は  $s$  方向に遮蔽境界を横切り易いという事実を反映している。

【0112】前に述べたように、正規化制約の役割は、光学流れ制約と方向平滑制約との重み付け平均に同じ最適値を与える助変数ベクトル場—387—の間を、これらの値になにも有意な変化を引き起こさずに識別することである。これは、連立非線形方程式(4-19)に現れる助変数  $\gamma$  と  $\gamma$  とに小さな値を割り当てることによって達成される。以下に述べる解析を簡単にするために、連立非線形方程式(4-19)の解が局所的には一意であるということを考慮しながら助変数  $\gamma$  と  $\gamma$  とはゼロに等しいと仮定することによって、正規化制約を無視する。

【0113】上記を考慮に入れて、次節で述べる連立非線形方程式(4-19)を解くプロセスは、二つの相反する制約の集合、すなわち光学流れ制約と方向平滑制約とを表す関数間で均等バランスを得ようとする試みと考えることができる。これは一部には、次節で述べる連立非線形方程式(4-19)の線形化(5-16)から生じる二次形式が正の定符号形式であるという事実の結果である。

29

【0114】関係(4-11)に現れる助変数 $r^2$ 、 $p^2$ 、 $q^2$ の値を大きくすることによって重み関数——388——の値を小さくすることによって理解するのは難しい。これは、連立非線形方程式(4-19)におけるバランスを方向平滑制約の方へシフトするであろう。光学流れ制約と方向平滑制約との間の均等バランスを回復するために、連立非線形方程式(4-19)は、方向平滑制約に対応する関数——389——の絶対値が減少する一方、光学流れ制約に対応する関数——390——の絶対値が増加するような仕方で反応するであろう。同様に、関係(4-13)に現れる助変数——391——の値を大きくすることによって重み関数——392——の値を小さくする。今度はこれは、連立非線形方程式(4-19)におけるバランスを光学流れ制約の方へシフトするであろう。光学流れ制約と方向平滑制約との間の均等バランスを回復するために、連立非線形方程式(4-19)は、光学流れ制約に対応する関数——393——の絶対値を減少させる一方、方向平滑制約に対応する関数——394——の絶対値を増加するような仕方で反応するであろう。

【0115】光学流れ制約と方向平滑制約との振る舞いに関する前記の論議を考慮に入れれば、次のように結論できる。助変数 $r^2$ と $a^2$ は、遮蔽境界から離れた画像点において連立非線形方程式(4-19)の解に最大の影響を与えるが、一方、助変数——395——は遮蔽境界の近傍にある画像点において連立非線形方程式(4-19)の解に最大の影響を与える。助変数 $r^2$ は、光学流れ制約を指定する関数——396——のための比例定数を定義するが、助変数 $a^2$ は、方向平滑制約指定する関数——397——のための比例定数を決定する。

【0116】助変数 $r^2$ と $p^2$ と $q^2$ の組み合わせ(4-24)は連立非線形方程式(4-19)の解に対する光学流れ制約の影響の上限を決定するが一方、助変数 $a^2$ と $c^2$ と——398——の組み合わせ(4-27)は連立非線形方程式(4-19)の解に対する方向平滑制約の影響の上限を決定する。

【0117】前記のように、助変数速度ベクトル場——400——の評価——399——は、連立非線形方程式(4-19)の解として定義された。これは、下記の形式——401——で表すことができる。ここで表記を簡潔にするために、連立非線形方程式(4-19)に現れる関数の引き数は省略されている。また、すべての画像点 $(x, y) \in \Omega$ において対応する関数は速度ベクトル——405——の評価——404——に依存するということを示すために、記号——403——の代わりに記号——402——が使われている。

【0118】助変数速度ベクトル場——406——の評価を計算するために、区間 $[1, \infty)$ からの助変数の有限部分集合 $\sigma_1, \dots, \sigma_n$ に対してわれわれ自身、ある制限を課している。

【0119】 $\sigma_n$ を $\infty$ に等しい値を持つ助変数とする。 50

30

助変数 $\sigma_n$ の値は1に等しいと仮定し、また助変数列 $\sigma_0, \sigma_1, \dots, \sigma_n$ はまばら過ぎることなく、値は減少して行くものと仮定する。それから、そこでは解は所定のベクトル定数——408——にまったく等しい助変数値 $\sigma = \sigma_0$ から始まり、そして先行する助変数値に関して得られた連立非線形方程式(5-1)の解を初期評価として用いる助変数値に関して減少していく方向に進みながら、助変数の各値 $\sigma = \sigma_1, \dots, \sigma_n$ に関して連立非線形方程式(5-1)を逐次解いて行くことにより、助変数速度ベクトル場——407——の評価が得られる。

【0120】 $\delta(\sigma)$ を下記の関係——409——によって助変数値 $\sigma = \sigma_1, \dots, \sigma_n$ の集合上で定義された助変数値 $\sigma$ の関数とする。

【0121】もし重み定数——410——が関係——411————412——のように定義されるならば、正規化制約——413——および——414——は、すべての画像点 $(x, y) \in \Omega$ とすべての助変数値 $\sigma = \sigma_1, \dots, \sigma_n$ に関してそれぞれ——415——

——416——のように近似することができる。これらの条件の下で助変数速度ベクトル場——417——の未知の評価の連立非線形方程式(5-1)は——418——という形式を取る。それから助変数速度ベクトル場を評価するための計算プロセスは、下記のようにさらに精密に述べるることができる。すなわち、すべての助変数値 $\sigma = \sigma_1, \dots, \sigma_n$ に関して、助変数値 $\delta(\sigma)$ に対応する速度ベクトル場のすでに得られた評価——419——を初期評価として使いながら、連立非線形方程式(5-7)を解くことにより、初期評価に最も近い助変数値 $\sigma$ に対応する速度ベクトル場の評価——420——を求める。

【0122】助変数 $\sigma = \sigma_1, \dots, \sigma_n$ のある固定値に関して、助変数値 $\sigma$ に対応する速度ベクトル場の未知の評価——421——に関する連立非線形方程式(5-7)は、すべての画像点 $(x, y) \in \Omega$ に関して、速度ベクトル——422——の成分と——423——という形式で表せるその偏導関数との間の関係の集合を定義する。初期評価——425——に最も近い助変数値 $\sigma$ に対応する連立非線形方程式(5-7)の解の評価として助変数値 $\sigma$ に対応する速度ベクトル場の評価——424——を求めるために、反復更新方式を用いることができる。この方式では、助変数値 $\sigma$ に対応する連立非線形方程式(5-7)の解の現行評価——427——として取られる初期評価——426——から始めて、助変数値 $\sigma$ に対応する連立非線形方程式(5-7)の解の改善された評価——428——が、ステップ——428-1——によって決定される。ここでスカラー助変数——428-2——はこのステップの長さを定義し、ベクトル場——429——はステップの方向を定義する。ステップ長——429-1——は下記の関数——

430——が最小になるような仕方で選択される。ステップ方向——431——は、関数(5-10)が $\omega \in (0, 1]$ の十分に小さい値に関して定常的に減少となるような仕方で選択される。改善された評価——432——は、助変数値 $\sigma$ に対応する連立非線形方程式(5-7)の解の現行評価——433——として得られ、また反復更新方式のステップ(5-9)は、助変数値 $\sigma$ に対応する連立非線形方程式(5-7)の解のその次の改善された評価——434——を決定するために繰り返される。このプロセスは、適当な基準が満たされるまで続けられる。その場合、改善された評価——435——は、助変数値 $\sigma$ に対応する速度ベクトル場の評価——436——として得られる。

【0123】もしベクトル場——437——が大き過ぎなければ、ベクトル場——438——の近傍で非線形作用素——439——は——440——のように線形に展開できる。ここでベクトル場——441——に関して線形でかつ有界の作用素——442——は、非線形作用素——443——のヤコビアンである。作用素——444——は、——445——という形式でさらに都合よく表現できる。

【0124】ニュートン法[22]においてはベクトル場——446——は連立線形方程式——447——の解として定義される。それから連立非線形方程式(5-7)の解の改善された評価——448——は、スカラー助変数——449——が1に等しく取られている関係(5-6)のように定義される。

【0125】連立線形方程式(5-13)を解き、そして関係(5-9)を適用した結果得られた連立非線形方程式(5-7)の解の改善された評価——450——は、必ずしもより良い評価とはならない。その背後にある理由は、非線形作用素——452——のヤコビアン——451——が非対象であって状態がよくないので連立線形方程式(5-13)がベクトル場——453——に関して信頼度よく解くことができないという事実による。最後は、正規化制約は関係(4-21)に現れる二次形式が正の定符号形式であるような遮蔽境界を越えている画像点において連立非線形方程式(5-7)の曖昧さを解決する場合にのみ有効であり得るという事実の結果である。正規化制約は、関係(4-21)に現れる二次形式が正の定符号形式でないような遮蔽境界の近傍にある画像点において連立非線形方程式(5-7)の曖昧さを解決する場合には無効になる。この観測は、代替の方法を採用すべきであるということを示唆している。

【0126】連立非線形方程式(5-7)を解く準ニュートン法においては、連立線形方程式(5-13)は連立線形方程式——454——で置き換えられる。この方法は、よく調整された方法の全域的収束戦略とニュートン法の迅速局所戦略とを、両者の利点を引き出すような仕方で組み合わせている。この準ニュートン法は助変数速度ベクトル場を評価する問題のために有意義な連立非線形方程式(5-7)の正規化を定義する。ベクトル場——455——

- に関して線形でかつ有界の作用素——456——は、次のように定義された非線形作用素——458——のヤコビアン——457——への近似である。

【0127】——459——は、関係——462——によって定義される助変数値 $\sigma$ に対応する速度ベクトル場の評価の不変コピ——460——と可変コピ——461——との非線形作用素の族であるとする。ここで記号——463——は、対応する関数はすべての点 $(x, y) \in \Omega$ に関して速度ベクトル——465——の評価の不変コピ——464——に依存するということを示すために使われるが、記号——466——は、対応する関数はすべての点 $(x, y) \in \Omega$ に関して速度ベクトル——468——の評価の可変コピ——467——に依存するということを示すために使われる。もし速度ベクトル場の評価の可変コピ——469——が速度ベクトル場の評価の不変コピ——470——に完全に等しいならば、すべての $\Theta \in [0, 1]$ に関して非線形作用素——471——は非線形作用素——472——に完全に等しい。助変数 $\Theta$ は、速度ベクトル場の評価の可変コピ——473——を介して光学流れ制約と方向平滑制約とのフィードバック緩和の程度を定義する。

【0128】——474——はベクトル場——475——に関して線形でかつ有界の作用素の族であるとする。ここでは各 $\Theta \in [0, 1]$ に関して作用素——476——は、ベクトル場——480——がベクトル場——481——と完全に等しいという条件の下で、ベクトル場——478——の関数と見なされかつ助変数に依存すると同じようにベクトル場——479——に依存している非線形作用素——477——のヤコビアンである。それから作用素——482——は、下記の要件を満足する族——483——のメンバーとして定義される。すなわち、

1. 連立線形方程式(5-14)の解として得られたベクトル場——484——は、全域的収束戦略の一部として、ベクトル場——488——がベクトル場——489——と完全に等しいという条件の下で、ベクトル場——486——の関数と見なされかつ助変数に依存すると同じようにベクトル場——487——に依存している汎関数——485——に対して下降方向であることが要求される。言い換えれば、スカラー変数 $\omega$ の関数と見なされる汎関数——490——の、変数 $\omega$ に関して第1階の偏導関数は、この変数 $\omega$ の値が0に等しいとき負であることが要求される。この導関数の値が——491——に等しいということを理解するのは難しくない。汎関数(5-16)が負であるための要件は、作用素——492——に正の定符号であることを要求することによっても満足することができる。

【0129】2. 迅速局所戦略の一部として作用素——493——は、非線形作用素——495——のヤコビアン——494——に可能な限り最も近い近似であることが要求される。

【0130】助変数値 $\Theta=0.5$ に対応する族からの作用素

----496---- は、上記の要件を満足し、また連立線形方程式 (5-14) に現れる作用素----497---- として得られる。作用素----498---- のこの選択によって、連立線形方程式 (5-14) は、下記の明示的形式----499---- を持つ。

【0131】次に、連立線形方程式 (5-17) のみならず連立非線形方程式 (5-7) の有限差分打ち切り法を考察する。連立非線形方程式 (5-7) の解の評価を構成する方法は、この連立方程式の有限差分打ち切り法に拡張される。

【0132】組合せ一般初期画像関数 $\Gamma$ 、 $g \in G$ は検定関数 $\Phi(R^3)$ の全集合上で定義されているが、速度ベクトル $(u(x, y, t), v(t))$ の評価----500---- はすべての画像点 $(x, y) \in \Omega$ とすべての視点 $t \in T$ に関して定義されているという場合には、助変数速度ベクトル場の評価の計算のための理論的解析は有効に行えるけれども、実際の計算は上記のデータセットの有限集合についてのみ遂行できる。本節では、連立非線形方程式 (5-7) の有限差分打ち切り法について述べる。助変数速度ベクトル場の評価は有限個の非線形方程式からなる連立式の解として決定される。

【0133】局所的凸型線形位相空間 $\Phi(R^3)$ 上で定義された一般関数 $F$ を与え、また検定関数 $\phi \in \Phi(R^3)$ を与えると、一般関数 $F$ と検定関数 $\phi$ との“合成”は、すべての $(x, y, t) \in R^3$ に関して関係----501----によって空間 $R^3$ 上で定義された無限に微分可能な関数 $(F * \phi)$ である。

【0134】 $F$ は一般関数であるとし、 $\chi \in \Phi(R^3)$ は----502---- というような所定の固定された負でない検定関数であるとする。例えば、----503---- を取る。ここで $\Psi$ は後述する関数である。検定関数 $\chi$ は“標本化関数”と呼ばれる。それから標本化関数 $\chi$ による一般関数 $F$ の“正規化”は、一般関数 $F$ と標本化関数 $\chi$ との合成として得られる無限に微分可能な関数 $(F * \chi)$ として空間 $R^3$ 上で定義される。

【0135】----504---- を視点列----505---- を含む増加視点列とする。すべての----506---- に関して、標本関数 $\chi$ を介して輝度画像関数 $\xi$ 、 $\xi \in \Xi$ に結び付けられた一般関数 $\Gamma_\xi$ の正規化----507---- と標本化関数 $\chi$ を介して特徴画像関数 $\eta$ 、 $\eta \in H$ に結び付けられた一般関数 $\Gamma_\eta$ の正規化 $\gamma \equiv (\Gamma * \chi)$ とは、画像平面の下記の部分集合----508----に属する点 $(x, y)$ において与えられる。ここで $Z$ は整数の集合、----509---- は二次元実数ベクトル、----510---- はこれらベクトルと整数係数----511---- との線形結合である。集合----512---- からの点は助変数 $\sigma$ に対応する“標本点”と呼ばれ、関数 $\gamma$ 、 $\xi \in \Xi$ は“標本化輝度画像関数”と呼ばれ、また関数 $\gamma$ 、 $\eta \in H$ は“標本化特徴画像関数”と呼ばれる。

【0136】標本化輝度画像関数----514---- に結び付

けられた一般関数----513---- と標本化特徴画像関数----516---- に結び付けられた一般関数----515---- とは検定関数 $\Phi(R^3)$ の集合上で次のように定義される。検定関数 $\phi \in \Phi(R^3)$ において標本化輝度画像関数----518---- に結び付けられた一般関数----517---- の値は関係----519----によって与えられる。検定関数 $\phi \in \Phi(R^3)$ において標本化特徴画像関数----521---- に結び付けられた一般関数----520---- の値は関係----522----によって与えられる。

10 【0137】----523---- は実数値定数の集合、 $g(\lambda)$ は集合 $\lambda$ に付けられたインデックス、そして----524---- は定数の集合 $\lambda$ に対応する負でない整数定数の集合であるとする。それからインデックス $g \equiv g(\lambda)$ に対応する“結合された一般標本化画像関数”は局所的凸型線形位相空間 $\Phi(R^3)$ 上で定義された一般関数----525---- である。検定関数 $\phi \in \Phi(R^3)$ における結合された一般標本化画像関数----526---- の値は、関係----527----によって与えられ、検定関数 $\phi$ における結合された一般標本化画像関数----528---- の“観測”と呼ばれる。

【0138】各集合----529---- は画像平面上の点の正方グリッドであるという場合に対する制限を仮定する。言い換えれば、これは、すべての $\sigma = \sigma_0, \sigma_1, \dots, \sigma_n$ に関して、ベクトル----530---- は長さが等しく、また互いに直交しているという要件である。またこれは、こういう点のグリッドは、下記の関係----531----が満足されることを意味するピラミッドを画像平面上に構成するという要件である。

30 【0139】“グリッド回転法”は上記の要件に従う標本化法である。これは、一般に偏導関数の数値解を得るために使われる。この方法では、ベクトル----532---- は互いに等しい長さでかつ直交しているように選択され、それからすべての $k=1, \dots, n$ に関してベクトル----533---- は関係----534---- ----535---- のように定義される。上記のように定義されたベクトル----536---- がすべての $k=0, 1, \dots, n$ に関して互いに等しい長さでかつ直交しており、また関係 (6-6) が満足されているということを理解するのは難しくない。

40 【0140】 $k'$ は正の整数定数であり、 $\Psi \in \Phi(R^3)$ は所定の測定関数であるとする。それからすべての視点----537---- とすべての助変数値 $\sigma = \sigma_0, \sigma_1, \dots, \sigma_n$ とに関して、画像関数----538---- とその空間偏導関数----539---- は集合 $\Omega$ 上で次のように定義される。

【0141】初めに、画像点 $(x, y)$ は部分集合----540---- に属していると仮定する。ここで二次元実数ベクトル----541---- は関係----542---- ----543---- のように定義される。一方、集合----544---- は関係----545---- のように定義される。それからす

すべての  $g \in G$  に関して画像関数——547——の成分——546——の値は、関係(2-4)によって定義される検定関数——550——における結合された一般標本画像関数——549——の観測——548——として決定され、また画像関数——553——の偏導関数——552——の成分——551——の値は、関係(2-5)、(2-6)によって定義される検定関数——556——における結合された一般標本画像関数——555——の観測——554——として与えられる。

【0142】  $(x, y)$  を集合  $\Omega$  からの任意の画像点とすると、間隔  $[0, 1]$  からの幾つかの整数  $i_1, i_2$  と幾つかの実数  $\theta_1, \theta_2$  とに関して下記の関係——557——が満足される。それから画像関数——558——の値とその偏導関数——559——の値は、関係——566——  
——567——  
——568——

——569——のように与えられる集合——565——からの点における画像関数——563——の値とそのそれぞれの偏導関数——564——の値との双線形補間——560——\*

$$(s, \nabla f(x, y, t, s)),$$

を持つ。ここで  $f(x, y, t, s)$  は、変数  $(x, y) \in \Omega, t \in T, s \in S$  の関数であって、この関数は変数  $t \in T, s \in S$  のすべての固定値について変数  $(x, y) \in \Omega$  に関して連続的に微分可能である。

【0145】 集合  $S$  を構成する単位円は次の性質を持つ集合——577-1——の、同じ記号  $S$  で表される有限集合で置き換えられる。すなわち、集合  $S$  は原点を含まず、また集合  $S$  に属するすべての元  $s$  に関して元  $-s$  もまた集合  $S$  に属す。測度  $ds$  はすべての元  $s \in S$  に値 0.5 を結び付ける点測度で置き換えられる。

【0146】 以上の作用の結果、形式(6-23)の任意の式の値の集合  $S$  についての積分は、その式の値の集合  $S$  についての有限になる。言い換えれば、次の関係——578——が満足される。

【0147】 すべての  $t \in T, s \in S$  について変数  $(x, y) \in \Omega$  に関して連続的に微分可能である関数  $f(x, y, t, s)$  と元  $s \in S$  とを与えると、式(6-23)は偏導関数への有限差分近似によって集合——579——上で——580——のように定義される。ここで——581——は助変数としてベクトル  $s$  の長さに依存する正の実数定数であり、また値——582——はベクトル——583——の成分である。

【0148】 関係(6-25)と、集合  $S$  はすべての元  $s \in S$  に関して元  $-s \in S$  であるという性質を持つという事実とを考慮に入れると、 $f(x, y, t, s)$  が変数  $s$  に関して奇関数である場合には関係(6-24)に現れる和は——584——のように表すことができる。

【0149】 集合  $G_i$  は有限であって、 $dg_i$  は値——585——を元——586——に結び付ける点測度であると仮定する。ここで——587——はすべての——588——

\*-

——561——

——562——によって得られる。

【0143】  $k$  は正の整数定数であるとし、——570——は関係——571——

——572——のように定義される二次元実数ベクトルであるとし、また集合——573——は関係——574——のように定義されるものとする。助変数値  $\sigma = \sigma_0, \sigma_1, \dots, \sigma_n$  に対応する両方の連立非線形方程式(5-7)と連立線形方程式(5-17)の有限差分打ち切り法は、助変数値  $\sigma$  に対応する速度ベクトル場——576——の未知の評価に関する連立非線形方程式およびその有限差分として、また助変数値  $\sigma$  に対応する未知のベクトル場——577——に関する連立線形方程式およびその有限差分として、集合——575——上でそれぞれ次のように定義される。

【0144】 連立非線形方程式(5-7)と連立線形方程式(5-17)とにおける集合  $S$  に対する積分記号の下に現れる式は、一般形式

$$(6-23)$$

に関して正の実数定数である。

【0150】 上記の論議に基づいて、連立非線形方程式(5-7)は集合——589——上で——590——のように定義できる。また一方、連立線形方程式(5-17)は集合——591——上で——592——のように定義できる。関係(6-27)、(6-28)に現れる関数の引き数は、表記を単純にするために省略されている。集合  $S$  に関する和分記号の下に現れる関数は、点——593——において評価されるが、残りの関数はすべての——594——に関して点  $(x, y, t)$  において評価される。関数——595——は、それぞれ関係

——596——

——597——

——598——によって定義される。関数——599——は関係——600——のように定義される。関数——601——は関係——602——によって定義されるが一方、関数——603——は関係——604——のように定義される。関数——605——は関係——606——によって定義される。ここで各関数——607——は——608——のように決定される。上記のように、各値  $(x, y, t)$  ( $(x, y, t)$  か——609——のいずれかに等しい)に関して関数——610——の値は関係(4-8)によって与えられる。

【0151】 集合——612——の集合——611——における連立非線形方程式(6-27)の制限と連立線形方程式(6-28)の制限は両者とも、下記のような境界条件の導入によって達成される。点——615——が集合——616——の外にあるようなすべての画像点——613——とすべての元——614——とについて、速度ベクトル——617——の評価の値は、速度ベクトル——618——評価の

37

値と完全に等しいように定義される。これらの条件の下で、連立非線形方程式 (6-27) と連立線形方程式 (6-28) とは、集合 619 からすべての画像点  $(x, y)$  に関してうまく定義されるので、これらの連立方程式は、助変数  $\sigma$  に対応する速度ベクトル場 620 の未知の評価に関する連立非線形方程式およびその有限差分と、助変数  $\sigma$  に対応する速度ベクトル場 621 の未知の評価に関する連立線形方程式およびその有限差分とを形成する。

【0152】集合  $S$  の元の数を記号  $|S|$  によって表すと、すべての 622 に関して定数 623 は関係 624 のように定義できる。

【0153】下記は集合  $S$  を選択するために一般に用いられる二つの方法 625

626 である。もし集合  $S$  が関係 (6-38) のように選択されれば、“5点有限差分打ち切り”が使われたと言い、また関係 (6-39) によって指定される選択については“9点有限差分打ち切り”が使われたと言う。

【0154】 $k=0, \dots, n-1$  とし、また 627 を助変数 628 に対応する速度ベクトル場の評価とする。そうすると、助変数 630 に対応する速度ベクトル場の初期評価 629 は次のように定義される。632 であるようなすべての 631 について評価 633 の値は、関係 634 によって与えられる。一方、636 であるようなすべての 635 について評価 637 の値は、関係 638 によって与えられる。ただし 639 である。

【0155】連立線形方程式 (6-28) は対称であってかつ正の定符号形式である。以下に示すように、連立線形方程式 (6-28) は、二つの系に分割でき、その一つは直接反転できる。これらの性質に基づいて、迅速事前調整反復法が構成される。もし集合 640 に、ある順序づけを導入すると、連立線形方程式 (6-28) は 641 のような行列表記で表される。ここで 642 は、指定された順序づけにしたがってまず集合 645 の元をリストニングし、それから元 646 をリストニングすることにより得られる、ベクトル場 644 に対応する行ベクトル 643 の転置行列ベクトルである。集合 647 についての“自然順序づけ”は次のように定義できる。関係 (6-2) と (6-22) は、すべての元 648 は  $i_1$  と  $i_2$  が整数である場合に 649 という形式で一意的に表現できるということを暗示している。そうすると、もし  $i_2 > i_1$  であるか、あるいはもし  $i_2 = i_1$  でかつ  $i_1 > i_1'$  であれば、元 650 は元 651 の次にくる。

【0156】関係 656 が満足されるように、行列 652 は行列 653 の対角部であるとしまた一方、行列 654 は行列 655 の非対

38

角部であるとする。そうすると連立線形方程式 (6-28) は 657 という形式を取る。

【0157】658 は関係 659 によって定義される対角行列であるとする、次の関係 660 が満足される。661 は関係 662 によって定義されるベクトルであるとする。もし未知のベクトル 663 の連立線形方程式 (7-3) の両辺が行列 664 上で左から乗算されるならば、そしてもし未知のベクトル 665 が未知のベクトル 666 で置きかえられるならば、連立線形方程式 (7-3) は未知のベクトル 667 に関して下記の連立線形方程式 668 となる。

【0158】669 は関係 670 のように定義される行列であるとし一方、671 は関係 672 によって与えられるベクトルであるとする、下記の基本的反復法を使って連立線形方程式 (6-28) の解に対する近似値を得ることができる。

【0159】連立線形方程式 (7-7) の解に対する初期近似値 673 はゼロにまったく等しいと定義される。すべての  $n=0, 1, \dots$  に関して近似値 674 は関係 676 のように近似値 675 によって定義される。このプロセスは、連立線形方程式 (7-7) の解に対する適当な近似値 677 得られるまで続けられる。連立線形方程式 (6-28) の解に対する近似値 678 は、関係 679 によって定義される。

【0160】基本的反復法 (7-10) の性能は、基本的反復法に適用される多項式加速法の助けによって改善できる。表現を簡単にするために、下記の表記 680 を使う。そうすると、基本的反復法 (7-10) は 681 となる。

【0161】基本的反復法 (7-13) の共役傾斜多項式加速法は、次のように述べる事ができる。ゼロにまったく等しい初期近似値のボールド  $w_0$  から始め、それからすべての  $n=0, 1, \dots$  に関して下記の反復手順 682 が適用される。ここで係数  $\rho$  と  $\gamma$  は関係 683 によって与えられる。但し 684 685 686 である。

【0162】共役傾斜法は、代替形式 687 で表すことができる。ここで係数  $\alpha$  は関係 (7-16) のように定義されが一方、ベクトル  $p$ 、 $q$  は関係 (7-17) によって与えられる。

【0163】本発明の例示的な実施例においては、逐次的高精細解像度レベル毎に2.0という因数だけ減少していく助変数  $\sigma$  の値を持った5レベル解像度ピラミッドが使われた。9点有限差分打ち切りが使われた。関係 (6-9)、(6-10) に現れる正の整数定数  $k'$  は4に等しく一方、関係 (6-20)、(6-21) に現れる正の整数定数  $k''$  は2に等しい。これらの値を用いて、本発明は多数の速

蔽境界を含む実世界シーンの画像の場合に良好な精度で速度ベクトル場を評価することができる。

【0164】本発明の多くの特徴と利点は、詳細な特許明細書から明らかであり、かくしてまた本発明に付属の特許請求の範囲は、本発明の真の意図と適用範囲とに入る本発明のこういった特徴と利点のすべてをカバーすることを意図している。さらに、当該技術に熟達した人々は容易に多数の修正並びに変更を思いつくであろうから、図示・説明された厳密な構成と操作とに本発明を限定することは望ましくなく、したがって適当な修正とそれ相当手段はすべて本発明の適用範囲に入るとすることができる。

【0165】なお、本明細書において——数字——は数式番号を示したものであって、数式番号と実際の数式の対応関係は、図1～図41に示してある。

#### 【図面の簡単な説明】

【図1】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図2】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図3】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図4】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図5】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図6】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図7】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図8】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図9】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図10】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図11】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図12】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図13】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図14】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図15】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図16】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図17】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図18】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図19】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図20】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図21】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図22】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図23】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図24】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図25】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図26】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図27】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図28】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図29】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図30】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図31】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図32】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図33】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図34】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図35】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図36】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図37】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図38】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図39】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図40】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図41】明細書中の数式番号と実際の数式の対応関係を示す図である。

【図42】本発明のハードウェア構成要素を描いているブロック図である。

【図43】対応処理の間の二次元探索を図解している図である。

【図44】対応処理の間の一次元探索を描いている図である。

【図45】本発明における処理のブロック図である。

【図46】デイスパリティ・ベクトル場処理を図解しているフローチャートである。

【図47】本発明を実施するためのフローチャートである。このフローチャートは、C言語で書かれたソースコードの付録マイクロフィッシュ。ここに含まれたマイクロフィッシュ付録はUnixオペレーティングシステムで動くサンマイクロシステムのSparc10コンピュータ上に本発明の一実施例を実現するのに適したソースコードを提供する。

【図48】本発明を実施するためのフローチャートである。

【図49】本発明を実施するためのフローチャートである。

【図50】本発明を実施するためのフローチャートである。

【図51】本発明を実施するためのフローチャートである。

【図52】本発明を実施するためのフローチャートである。

【図53】本発明を実施するためのフローチャートである。

【図54】本発明を実施するためのフローチャートである。

【図55】本発明を実施するためのフローチャートである。

【図56】本発明を実施するためのフローチャートである。

【図57】本発明を実施するためのフローチャートである。

【図58】本発明を実施するためのフローチャートである。

【図59】本発明を実施するためのフローチャートである。

【図60】本発明を実施するためのフローチャートである。

【図61】本発明を実施するためのフローチャートである。

【図62】本発明を実施するためのフローチャートである。

【図63】本発明を実施するためのフローチャートである。

【図64】本発明を実施するためのフローチャートである。

【図65】本発明を実施するためのフローチャートである。

【図66】本発明を実施するためのフローチャートである。

【図67】本発明を実施するためのフローチャートである。

【図68】本発明を実施するためのフローチャートである。

【図69】本発明を実施するためのフローチャートである。

【図70】本発明を実施するためのフローチャートである。

【図71】本発明を実施するためのフローチャートである。

【図72】本発明を実施するためのフローチャートである。

【図73】本発明を実施するためのフローチャートである。

【図74】本発明を実施するためのフローチャートである。

【図75】本発明を実施するためのフローチャートである。

【図76】本発明を実施するためのフローチャートである。

【図77】本発明を実施するためのフローチャートである。

【図78】本発明を実施するためのフローチャートである。

【図79】本発明を実施するためのフローチャートである。

【図80】本発明を実施するためのフローチャートである。

【図81】本発明を実施するためのフローチャートである。

【図82】本発明を実施するためのフローチャートである。

【図83】本発明を実施するためのフローチャートである。

【図84】本発明を実施するためのフローチャートである。

【図85】本発明を実施するためのフローチャートである。

【図86】本発明を実施するためのフローチャートである。

【図87】本発明を実施するためのフローチャートである。

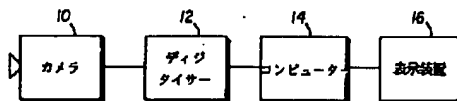
【図88】本発明を実施するためのフローチャートである。

【図89】本発明を実施するためのフローチャートである。

【図90】本発明を実施するためのフローチャートである。

【図 9 1】本発明を実施するためのフローチャートである。  
 【図 9 2】本発明を実施するためのフローチャートである。  
 【図 9 3】本発明を実施するためのフローチャートである。  
 【図 9 4】本発明を実施するためのフローチャートである。  
 【図 9 5】本発明を実施するためのフローチャートである。  
 【図 9 6】本発明を実施するためのフローチャートである。  
 【図 9 7】本発明を実施するためのフローチャートである。  
 【図 9 8】本発明を実施するためのフローチャートである。  
 【図 9 9】本発明を実施するためのフローチャートである。  
 【図 1 0 0】本発明を実施するためのフローチャートである。  
 【図 1 0 1】本発明を実施するためのフローチャートである。  
 【図 1 0 2】本発明を実施するためのフローチャートである。  
 【図 1 0 3】本発明を実施するためのフローチャートである。  
 【図 1 0 4】本発明を実施するためのフローチャートである。  
 【図 1 0 5】本発明を実施するためのフローチャートである。  
 【図 1 0 6】本発明を実施するためのフローチャートである。  
 【図 1 0 7】本発明を実施するためのフローチャートである。  
 【図 1 0 8】本発明を実施するためのフローチャートである。

【図 4 2】

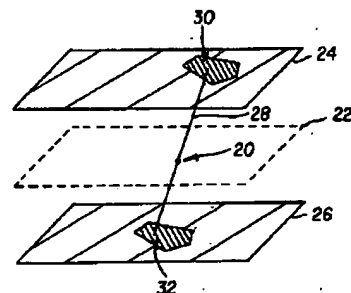


ある。

## 【符号の説明】

- 10 カメラ
- 12 デジタイザー
- 14 コンピュータ
- 16 表示装置
- 20 画素
- 22 中間画像
- 24、26 画像
- 28 ベクトルまたは線
- 30、32 二次元領域
- 38 水平軸
- 50 右デジタル画像
- 52 左デジタル画像
- 54 左右画像間の対応を確定せよ
- 56、58 中間ディスパリティ・ベクトル場を評価せよ
- 60 中間画像の補間とインターレース
- 62 奥行き画像
- 70 中間ディスパリティ・ベクトル場の初期評価を選択せよ
- 72 フィルタリングされた画像バンドの値とフィルタリングされた画像バンドの偏導関数の値を計算せよ
- 74 中間ディスパリティ・ベクトル場の初期評価を取得せよ
- 76 連立線形方程式を形成せよ
- 78 連立線形方程式を解け
- 80 所望の程度に改善された中間ディスパリティ・ベクトル場の現行評価
- 82 最高解像度の現行レベル
- 84 多段解像度ピラミッドの現行レベルから次段高解像度レベルへの昇段、および中間ディスパリティ・ベクトル場の終結評価の現行高解像度レベルの初期評価への投影

【図 4 3】



【図1】

- 1 :  $\{(\tilde{u}^\sigma(x, y, t_k), \tilde{u}(t_k)) \mid (x, y) \in \Omega, \sigma \in [1, \infty]\}$
- 2 :  $\{t''_0, \dots, t''_{k''}\}$
- 3 :  $\Gamma_\xi(\phi) = \iiint_{R^3} \xi(x, y, t) \phi(x, y, t) dx dy dt. \quad (2-1)$
- 4 :  $(\Gamma_\xi \mid \xi \in \Xi)$
- 5 :  $M(t) \subset \mathbb{R}^2$
- 6 :  $(x_\mu, y_\mu, t_\mu) \in M(t)$
- 7 :  $M \equiv \bigcup_{t \in \mathbb{R}} M(t)$
- 8 :  $B \equiv B(M)$
- 9 :  $\Gamma_\eta(\phi) = \iiint_M \eta(x_\mu, y_\mu, t_\mu) \phi(x_\mu, y_\mu, t_\mu) d\mu \quad (2-2)$
- 10 :  $(\Gamma_\eta \mid \eta \in H)$
- 11 :  $\frac{\partial^{m_x+m_y+m_t}}{\partial x^{m_x} \partial y^{m_y} \partial t^{m_t}}$
- 12 :  $\frac{\partial^{m_x+m_y+m_t}}{\partial x^{m_x} \partial y^{m_y} \partial t^{m_t}} F$
- 13 :  $\phi_{m_x, m_y, m_t} \in \Phi(R^3), \phi_{m_x, m_y, m_t}(x, y, t) =$   
 $(-1)^{m_x+m_y+m_t} \frac{\partial^{m_x+m_y+m_t}}{\partial x^{m_x} \partial y^{m_y} \partial t^{m_t}} \phi(x, y, t), (x, y, t) \in R^3.$
- 14 :  $\lambda \equiv \{\lambda_\xi, \lambda_\eta \mid \xi \in \Xi, \eta \in H\}$
- 15 :  $m_{g(\lambda)} \equiv \{m_{\lambda, x}, m_{\lambda, y}, m_{\lambda, t}, m_{\eta, x}, m_{\eta, y}, m_{\eta, t} \mid \xi \in \Xi, \eta \in H\}$
- 16 :  $\Gamma_{g(\lambda)}(\phi) = \sum_{\xi \in \Xi} \lambda_\xi \frac{\partial^{m_{\lambda, x}+m_{\lambda, y}+m_{\lambda, t}}}{\partial x^{m_{\lambda, x}} \partial y^{m_{\lambda, y}} \partial t^{m_{\lambda, t}}} \Gamma_\xi(\phi) + \sum_{\eta \in H} \lambda_\eta \frac{\partial^{m_{\eta, x}+m_{\eta, y}+m_{\eta, t}}}{\partial x^{m_{\eta, x}} \partial y^{m_{\eta, y}} \partial t^{m_{\eta, t}}} \Gamma_\eta(\phi)$   
(2-3)
- 17 :  $(\Gamma_{g(\lambda)} \mid \lambda \in \Lambda)$

【図2】

$$18: \psi \in \Phi(R^3)$$

$$19: \Omega \subset R^2$$

$$20: \sigma \in [1, \infty)$$

$$21: g^\sigma(x, y, t)$$

$$22: \psi_{x, y, t}^\sigma \in \Phi(R^3)$$

$$23: \psi_{x, y, t}^\sigma(\tilde{x}, \tilde{y}, \tilde{t}) = \frac{1}{\sigma^3} \psi((\tilde{x}-x)/\sigma, (\tilde{y}-y)/\sigma, (\tilde{t}-t)/\sigma), \\ (\tilde{x}, \tilde{y}, \tilde{t}) \in R^3. \quad (2-4)$$

$$24: \Gamma_g(\psi_{x, y, t}^\sigma)$$

$$25: g^\sigma$$

$$26: [1, \infty)$$

$$27: \psi_{x, y, t}^\sigma \in \Phi(R^3)$$

$$28: \Gamma_g(\psi_{x, y, t}^\sigma)$$

$$29: g^\sigma(x, y, t), g \in G$$

$$30: g^\sigma(x, y, t) = \{g^\sigma(x, y, t) \mid g \in G\}, (x, y, t) \in \Omega \times T, \sigma \in [1, \infty)$$

$$31: \sigma \in [1, \infty)$$

$$32: g^\sigma(x, y, t), (x, y, t) \in \Omega \times T$$

$$33: \{g^\sigma(x, y, t) \mid (x, y) \in \Omega\}$$

$$34: \sigma \in [1, \infty)$$

$$35: \{g^\sigma(x, y, t) \mid (x, y) \in \Omega\}$$

$$36: \{g^\sigma(x, y, t) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$37: g^\sigma(x, y, t)$$

$$38: g^\sigma(x, y, t), g \in G$$

【図3】

$$39: \psi_{x,y,t}^{\sigma} \in \Phi(R^3)$$

$$40: g^{\sigma}(x,y,t)$$

$$41: g_x^{\sigma}(x,y,t) = \{g_x^{\sigma}(x,y,t) | g \in G\}, g_y^{\sigma}(x,y,t) = \{g_y^{\sigma}(x,y,t) | g \in G\}$$

$$42: g_x^{\sigma}(x,y,t), g_y^{\sigma}(x,y,t), g \in G$$

$$43: (x,y,t) \in \Omega \times T, \sigma \in [1, \infty)$$

$$44: \frac{\partial}{\partial x} \psi_{x,y,t}^{\sigma}, \frac{\partial}{\partial y} \psi_{x,y,t}^{\sigma} \in \Phi(R^3)$$

$$45: \Gamma_g(\frac{\partial}{\partial x} \psi_{x,y,t}^{\sigma}), \Gamma_g(\frac{\partial}{\partial y} \psi_{x,y,t}^{\sigma})$$

$$46: \frac{\partial}{\partial x} \psi_{x,y,t}^{\sigma}(\bar{x}, \bar{y}, \bar{t}) = -\frac{1}{\sigma^4} \psi_x((\bar{x}-x)/\sigma, (\bar{y}-y)/\sigma, (\bar{t}-t)/\sigma), (\bar{x}, \bar{y}, \bar{t}) \in R^3, \quad (2-5)$$

$$47: \frac{\partial}{\partial y} \psi_{x,y,t}^{\sigma}(\bar{x}, \bar{y}, \bar{t}) = -\frac{1}{\sigma^4} \psi_y((\bar{x}-x)/\sigma, (\bar{y}-y)/\sigma, (\bar{t}-t)/\sigma), (\bar{x}, \bar{y}, \bar{t}) \in R^3, \quad (2-6)$$

$$48: t - \Delta t^-, t, t + \Delta t^+ \in R$$

$$49: (x(t - \Delta t^-), y(t - \Delta t^-)) \in R^2$$

$$50: (x(t + \Delta t^+), y(t + \Delta t^+)) \in R^2$$

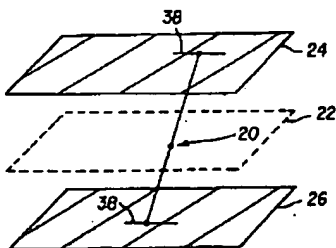
$$51: (x, y) \in R^2$$

$$52: (x, y) \in R^2$$

$$53: t \in R$$

$$54: (x, y) \in \Omega(t, \Delta t^-, \Delta t^+)$$

【図44】



【図4】

$$55: W(x, y, t, \Delta t) = \bigcup_{\substack{0 < \Delta t^- < \Delta t \\ 0 < \Delta t^+ < \Delta t}} W(x, y, t, \Delta t^-, \Delta t^+), \quad (2-11)$$

$$56: W(x, y, t) = \bigcap_{0 < \Delta t < \delta t} \overline{W(x, y, t, \Delta t)}, \quad (2-12)$$

$$57: \overline{W(x, y, t, \Delta t)}$$

$$58: u_I(x, y, t), v_I(x, y, t), (x, y, t) \in R^3$$

$$59: U_I(\Phi) = \iiint_{R^3} u_I(x, y, t) \Phi(x, y, t) dx dy dt, \quad (2-13)$$

$$60: V_I(\Phi) = \iiint_{R^3} v_I(x, y, t) \Phi(x, y, t) dx dy dt, \quad (2-14)$$

$$61: [1, \infty)$$

$$62: (u^\sigma(x, y, t), v^\sigma(x, y, t))$$

$$63: u^\sigma(x, y, t) = \frac{1}{\sigma^3} \iiint_{R^3} u_I(\tilde{x}, \tilde{y}, \tilde{t}) \Psi((\tilde{x}-x)/\sigma, (\tilde{y}-y)/\sigma, (\tilde{t}-t)/\sigma) d\tilde{x} d\tilde{y} d\tilde{t}, \quad (2-15)$$

$$64: v^\sigma(x, y, t) = \frac{1}{\sigma^3} \iiint_{R^3} v_I(\tilde{x}, \tilde{y}, \tilde{t}) \Psi((\tilde{x}-x)/\sigma, (\tilde{y}-y)/\sigma, (\tilde{t}-t)/\sigma) d\tilde{x} d\tilde{y} d\tilde{t}, \quad (2-16)$$

$$65: (u^\sigma(x, y, t), v^\sigma(x, y, t))$$

$$66: v^\sigma(x, y, t)$$

$$67: v^\sigma(t)$$

$$68: (u^\sigma(x, y, t), v^\sigma(t))$$

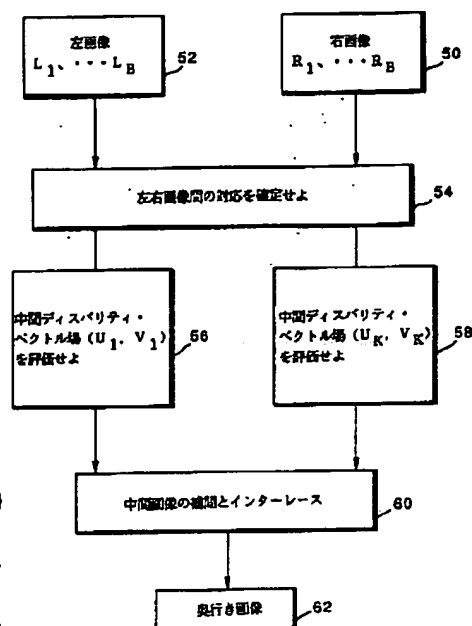
$$69: ((u^\sigma(x, y, t), v^\sigma(t)) | (x, y) \in \Omega)$$

$$70: (g^\sigma(x, y, t) | (x, y) \in \Omega)$$

【図5】

- 71:  $\{g^{\sigma}(x, y, t) \mid (x, y) \in \Omega\}$   
 72:  $\{(\{u^{\sigma}(x, y, t), v^{\sigma}(t)\} \mid (x, y) \in \Omega)\}$   
 73:  $\{(\{u^{\sigma}(x, y, t), v^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 74:  $\{g^{\sigma}(x, y, t) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$   
 75:  $t_0^* < t_1^* < \dots < t_k^*$   
 76:  $t_k^*, k=0, 1, \dots, K^*$   
 77:  $\{g^{\sigma}(x, y, t_k^*) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}_{k=0}^{K^*}$   
 78:  $\{g^{\sigma}(x, y, t) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$   
 79:  $\{(\{u^{\sigma}(x, y, t), v^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 80:  $\{(\{\tilde{u}^{\sigma}(x, y, t), \tilde{v}^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 81:  $\{(\{u^{\sigma}(x, y, t), v^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 82:  $\{g^{\sigma}(x, y, t_k^*) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}_{k=0}^{K^*}$   
 83:  $\{g^{\sigma}(x, y, t_k^*) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}_{k=0}^{K^*}$   
 84:  $t_k^*, k=0, 1, \dots, K^*$   
 85:  $\{(\{u^{\sigma}(x, y, t), v^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 86:  $\{(\{\tilde{u}^{\sigma}(x, y, t), \tilde{v}^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 87:  $\{g^{\sigma}(x, y, t_k^*) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}_{k=0}^{K^*}$   
 88:  $\{(\{u^{\sigma}(x, y, t), v^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 89:  $\{(\{\tilde{u}^{\sigma}(x, y, t), \tilde{v}^{\sigma}(t)\} \mid (x, y) \in \Omega, \sigma \in [1, \infty))\}$   
 90:  $\eta(x_{\mu}, y_{\mu}, t_{\mu}), \eta \in H$   
 91:  $(x_{\mu}, y_{\mu}, t_{\mu}) \in M$

【図45】



【図6】

$$92: \Pi_t \equiv \{(\Delta t^-, \Delta t^+) \mid (\Delta t^-, \Delta t^+) \in \mathbb{R}^2, \Delta t^- + \Delta t^+ > 0, \\ (t - \Delta t^-), (t + \Delta t^+) \in (t'')_{k=0}^{\infty}\} \quad (3-1)$$

$$93: G_t \equiv G \times P_t$$

$$94: g_t \equiv (g, \Delta t^-, \Delta t^+) \in G_t, \sigma \in [1, \infty)$$

$$95: g_t^\sigma(x, y, t, u^\sigma, v^\sigma)$$

$$96: g_t^\sigma(x, y, t, u^\sigma, v^\sigma) \equiv (g^\sigma(x + \Delta t^- u^\sigma(x, y, t), y + \Delta t^+ v^\sigma(t), t + \Delta t^+) \\ - g^\sigma(x - \Delta t^- u^\sigma(x, y, t), y - \Delta t^- v^\sigma(t), t - \Delta t^-)) / (\Delta t^- + \Delta t^+), \quad (3-2)$$

$$97: (u^\sigma, v^\sigma) \equiv (u^\sigma(x, y, t), v^\sigma(t))$$

$$98: g_t^\sigma$$

$$99: g_t^\sigma(x, y, t, u^\sigma, v^\sigma)$$

$$100: (u^\sigma(x, y, t), v^\sigma(t))$$

$$101: (\tilde{u}^\sigma, \tilde{v}^\sigma) \equiv (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t))$$

$$102: (g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2 \quad (3-3)$$

$$103: \tilde{u}^\sigma \equiv \tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma \equiv \tilde{v}^\sigma(t)$$

$$104: g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$105: (\Delta t^-)^{n-1} + (\Delta t^+)^{n-1}$$

$$106: g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$107: \tilde{u}^\sigma, \tilde{v}^\sigma$$

$$108: (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t))$$

$$109: \begin{cases} g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) = 0, \\ g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) = 0, \end{cases} \quad (3-4)$$

【図7】

$$110: g_{td}^o(x, y, t, \tilde{u}^o, \tilde{v}^o)$$

$$111: g_{tv}^o(x, y, t, \tilde{u}^o, \tilde{v}^o)$$

$$112: \tilde{u}^o$$

$$113: \tilde{v}^o$$

$$114: g_{\tau}^o(x, y, t, \tilde{u}^o, \tilde{v}^o)$$

$$115: (\Delta u^o(x, y, t), \Delta v^o(t))$$

$$116: \tilde{u}^o(x, y, t) = u^o(x, y, t) + \Delta u^o(x, y, t), \quad (3-5)$$

$$117: \tilde{v}^o(t) = v^o(t) + \Delta v^o(t). \quad (3-6)$$

$$118: (u^o(x, y, t), v^o(t))$$

$$119: u^o = u^o(x, y, t)$$

$$120: v^o = v^o(t)$$

$$121: g_{\tau}^o(x, y, t, u^o, v^o)$$

$$122: (\Delta u^o(x, y, t), \Delta v^o(t))$$

$$123: \begin{cases} (g_{tu}^o(x, y, t, u^o, v^o))^2 \Delta u^o(x, y, t) + g_{tu}^o(x, y, t, u^o, v^o) \\ \quad g_{tv}^o(x, y, t, u^o, v^o) \Delta v^o(t) + g_{tu}^o(x, y, t, u^o, v^o) \\ \quad g_{\tau}^o(x, y, t, u^o, v^o) = 0, \\ g_{tu}^o(x, y, t, u^o, v^o) g_{tv}^o(x, y, t, u^o, v^o) \Delta u^o(x, y, t) + \\ \quad (g_{tv}^o(x, y, t, u^o, v^o))^2 \Delta v^o(t) + g_{tv}^o(x, y, t, u^o, v^o) \\ \quad g_{\tau}^o(x, y, t, u^o, v^o) = 0. \end{cases} \quad (3-7)$$

$$124: g_{tu}^o(x, y, t, u^o, v^o)$$

$$125: g_{tv}^o(x, y, t, u^o, v^o)$$

$$126: u^o$$

$$127: v^o$$

$$128: g_{\tau}^o(x, y, t, u^o, v^o)$$

【図8】

$$129: u^{\sigma}$$

$$130: v^{\sigma}$$

$$131: g_t^{\sigma}(x, y, t, u^{\sigma}, v^{\sigma})$$

$$132: (g_{tu}^{\sigma}(x, y, t, u^{\sigma}, v^{\sigma}), g_{tv}^{\sigma}(x, y, t, u^{\sigma}, v^{\sigma})) \quad (3-8)$$

$$133: g_{tu}^{\sigma}(x, y, t, u^{\sigma}, v^{\sigma}) \Delta u^{\sigma}(x, y, t) + g_{tv}^{\sigma}(x, y, t, u^{\sigma}, v^{\sigma}) \Delta v^{\sigma}(t) + g_t^{\sigma}(x, y, t, u^{\sigma}, v^{\sigma}) = 0. \quad (3-9)$$

$$134: (u^{\sigma}(x, y, t), v^{\sigma}(t))$$

$$135: (\tilde{u}^{\sigma}(x, y, t), \tilde{v}^{\sigma}(t))$$

$$136: (u^{\sigma}(x, y, t), v^{\sigma}(t))$$

$$137: (\tilde{u}^{\sigma}(x, y, t), \tilde{v}^{\sigma}(t))$$

$$138: \{(u^{\sigma}(x, y, t), v^{\sigma}(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$139: \{(\tilde{u}^{\sigma}(x, y, t), \tilde{v}^{\sigma}(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$140: S = \{(s_x, s_y) \mid (s_x, s_y) \in \mathbb{R}^2, s_x^2 + s_y^2 = 1\}$$

$$141: s = (s_x, s_y) \in S$$

$$142: s = (s_x, s_y) \in S$$

$$143: (s, \nabla u^{\sigma}(x, y, t))$$

$$144: (s, \nabla u^{\sigma}(x, y, t)) = \left( \frac{\partial}{\partial \omega} u^{\sigma}(x + \omega s_x, y + \omega s_y, t) \right)_{\omega=0} \quad (3-10)$$

$$145: s = (s_x, s_y) \in S$$

$$146: (x + \omega s_x, y + \omega s_y)$$

$$147: (s, \nabla u^{\sigma}(x, y, t))$$

$$148: (u^{\sigma}(x, y, t), v^{\sigma}(t))$$

【図9】

$$149: (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t))$$

$$150: (s, \nabla \tilde{u}^\sigma(x, y, t))^2 \quad (3-11)$$

$$151: \{ (u^\sigma(x, y, t), v^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty) \}$$

$$152: \{ (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty) \}$$

$$153: (u^\sigma(x, y, t), v^\sigma(t))$$

$$154: (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t))$$

$$155: (\tilde{u}_0^\sigma(x, y, t), \tilde{v}_0^\sigma(t))$$

$$156: (u^\sigma(x, y, t), v^\sigma(t))$$

$$157: (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t))$$

$$158: (\tilde{u}^\sigma(x, y, t) - \tilde{u}_0^\sigma(x, y, t))^2 \quad (3-12)$$

$$159: (\tilde{v}^\sigma(t) - \tilde{v}_0^\sigma(t))^2 \quad (3-13)$$

$$160: (\tilde{u}_0^\sigma(x, y, t), \tilde{v}_0^\sigma(t))$$

$$161: (u^\sigma(x, y, t), v^\sigma(t))$$

$$162: (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t))$$

$$163: \{ (u^\sigma(x, y, t), v^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty) \}$$

$$164: \{ (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty) \}$$

$$165: (\tilde{u}, \tilde{v}) = \{ (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty) \}$$

$$166: (\tilde{u}, \tilde{v}) = \{ (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty) \}$$

$$167: (\Delta \tilde{u}, \Delta \tilde{v}) = \{ (\Delta \tilde{u}^\sigma(x, y, t), \Delta \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty) \}$$

【図10】

$$\begin{aligned}
 168: & (\Delta \bar{u}^\sigma(x, y, t), \Delta \bar{v}^\sigma(t)) \\
 & = (\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) - (\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)), \\
 & (x, y) \in \Omega, \sigma \in [1, \infty). \quad (4-1)
 \end{aligned}$$

$$169: \partial \Omega$$

$$170: (\Delta \bar{u}, \Delta \bar{v})$$

$$171: \partial \Omega$$

$$172: f(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$173:$$

$$\begin{aligned}
 f(\bar{u}, \bar{v}, \bar{u}, \bar{v}) = & \int_1^{\infty} \left[ \int_0^{\infty} \left[ \int_0^{\infty} 0.5 \alpha_{\sigma}^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, |\nabla \bar{u}^\sigma|) \right. \right. \\
 & (g_c^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2 dg_c + \\
 & \int_0^{\infty} 0.5 \beta_s^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, (s, \nabla \bar{u}^\sigma)) (s, \nabla \bar{u}^\sigma(x, y, t))^2 ds + \\
 & \left. \int_0^{\infty} 0.5 \gamma_{u,0}(\bar{u}^\sigma(x, y, t) - \bar{u}_0^\sigma(x, y, t))^2 dx dy + 0.5 \gamma_{v,0} \right. \\
 & \left. (\bar{v}^\sigma(t) - \bar{v}_0^\sigma(t))^2 \right] d\sigma. \quad (4-2)
 \end{aligned}$$

$$174: (\bar{u}, \bar{v}), (\bar{u}, \bar{v})$$

$$175: \alpha_{\sigma}^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, |\nabla \bar{u}^\sigma|), (x, y) \in \Omega, \sigma \in [1, \infty), g_c \in G_c$$

$$176: \bar{u}^\sigma = \bar{u}^\sigma(x, y, t), \bar{v}^\sigma = \bar{v}^\sigma(t), |\nabla \bar{u}^\sigma| = |\nabla \bar{u}^\sigma(x, y, t)|$$

$$177: \beta_s^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, (s, \nabla \bar{u}^\sigma)), (x, y) \in \Omega, \sigma \in [1, \infty), s \in S$$

$$178: \bar{u}^\sigma = \bar{u}^\sigma(x, y, t), \bar{v}^\sigma = \bar{v}^\sigma(t), (s, \nabla \bar{u}^\sigma) = (s, \nabla \bar{u}^\sigma(x, y, t))$$

$$179: |\nabla \bar{u}^\sigma(x, y, t)|, (x, y) \in \Omega, \sigma \in [1, \infty)$$

$$180: \nabla \bar{u}^\sigma(x, y, t)$$

【図11】

$$181: |\nabla \bar{u}^\sigma(x, y, t)| = \sqrt{\bar{u}_x^\sigma(x, y, t)^2 + \bar{u}_y^\sigma(x, y, t)^2}, \quad (4-3)$$

$$182: \bar{u}^\sigma(x, y, t)$$

$$183: (s, \nabla \bar{u}^\sigma(x, y, t)), (s, \nabla \bar{u}^\sigma(x, y, t)), (x, y) \in \Omega, \sigma \in [1, \infty), s \in S$$

$$184: g_\tau^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma), (x, y) \in \Omega, \sigma \in [1, \infty), g_\tau \in G_\tau$$

$$185: \bar{u}_0^\sigma(x, y, t), \bar{v}_0^\sigma(t)$$

$$186: (\bar{u}, \bar{v})$$

$$187: (\bar{u}, \bar{v})$$

$$188: (\bar{u}, \bar{v})$$

$$189: (\bar{u}, \bar{v})$$

$$190: f(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$191: (\bar{u}, \bar{v})$$

$$192: f(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$193: f(\bar{u}, \bar{v}, \bar{u} + \Delta \bar{u}, \bar{v} + \Delta \bar{v})$$

$$194: (\Delta \bar{u}, \Delta \bar{v})$$

$$195: (\bar{u}, \bar{v})$$

$$196: f(\bar{u}, \bar{v}, \bar{u} + \Delta \bar{u}, \bar{v} + \Delta \bar{v})$$

$$197: (\bar{u}, \bar{v})$$

$$198: (\Delta \bar{u}, \Delta \bar{v})$$

$$199: (\bar{u}, \bar{v})$$

$$200: (\Delta \bar{u}, \Delta \bar{v})$$

【図12】

201:  $\mathbf{f}(\bar{\mathbf{u}}, \bar{\mathbf{v}}, \bar{\mathbf{u}} + \Delta \bar{\mathbf{u}}, \bar{\mathbf{v}} + \Delta \bar{\mathbf{v}})$ 202:  $(\Delta \bar{\mathbf{u}}, \Delta \bar{\mathbf{v}})$ 203:  $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ 

204: 
$$\int_{G_c} \alpha_{g_c}^o(x, y, t, \bar{\mathbf{u}}^o, \bar{\mathbf{v}}^o, |\nabla \bar{\mathbf{u}}^o|) g_{c0}^o(x, y, t, \bar{\mathbf{u}}^o, \bar{\mathbf{v}}^o) g_c^o(x, y, t, \bar{\mathbf{u}}^o, \bar{\mathbf{v}}^o) dg_c -$$

$$\int (s, \nabla(\beta_s^o(x, y, t, \bar{\mathbf{u}}^o, \bar{\mathbf{v}}^o, (s, \nabla \bar{\mathbf{u}}^o)(s, \nabla \bar{\mathbf{u}}^o(x, y, t)))) ds +$$

$$\gamma_{u,0}(\bar{\mathbf{u}}^o(x, y, t) - \bar{\mathbf{u}}_0^o(x, y, t)) = 0, \quad (4-4)$$

205: 
$$\iint_{\Omega_{G_c}} \alpha_{g_c}^o(x, y, t, \bar{\mathbf{u}}^o, \bar{\mathbf{v}}^o, |\nabla \bar{\mathbf{u}}^o|) g_{c0}^o(x, y, t, \bar{\mathbf{u}}^o, \bar{\mathbf{v}}^o) g_c^o(x, y, t, \bar{\mathbf{u}}^o, \bar{\mathbf{v}}^o) dg_c dx dy +$$

$$\gamma_{v,0}(\bar{\mathbf{v}}^o(t) - \bar{\mathbf{v}}_0^o(t)) = 0,$$

206: 欠番

207: 欠番

208: 欠番

209: 欠番

210: 欠番

211: 欠番

212: 欠番

213: 欠番

214: 欠番

215: 欠番

【図13】

216: 欠番

217: 欠番

218: 欠番

219: 欠番

220:  $g_{t0}^{\sigma}(x, y, t, \bar{u}^{\sigma}, \bar{v}^{\sigma})$ 221:  $g_{t0}^{\sigma}(x, y, t, \bar{u}^{\sigma}, \bar{v}^{\sigma})$ 222:  $g_t^{\sigma}(x, y, t, \bar{u}^{\sigma}, \bar{v}^{\sigma})$ 223:  $\bar{u}^{\sigma} = \bar{u}^{\sigma}(x, y, t)$ 224:  $\bar{v}^{\sigma} = \bar{v}^{\sigma}(t)$ 225:  $f(\bar{u}, \bar{v}, \bar{u}, \bar{v})$ 226:  $(\bar{u}, \bar{v})$ 227:  $(\bar{u}, \bar{v})$ 228:  $\{(\bar{u}^{\sigma}(x, y, t), \bar{v}^{\sigma}(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$ 229:  $f(\bar{u}, \bar{v}, \bar{u}, \bar{v})$ 

230: 
$$f(\bar{u}, \bar{v}, \bar{u}, \bar{v}) = \int_1^{\infty} \left[ \iint_{\Omega} \left[ \int_{\Omega_t} 0.5 \alpha_{g_t}^{\sigma}(x, y, t, \bar{u}^{\sigma}, \bar{v}^{\sigma}, |\nabla \bar{u}^{\sigma}|) \right. \right. \\ \left. \left. (g_t^{\sigma}(x, y, t, \bar{u}^{\sigma}, \bar{v}^{\sigma}))^2 dg_t + \right. \right. \\ \left. \left. \int_{\Omega} 0.5 \beta_s^{\sigma}(x, y, t, \bar{u}^{\sigma}, \bar{v}^{\sigma}, (s, \nabla \bar{u}^{\sigma})) \right. \right. \\ \left. \left. (s, \nabla \bar{u}^{\sigma}(x, y, t))^2 ds + \right. \right. \\ \left. \left. 0.5 \gamma_{u,0}(\bar{u}^{\sigma}(x, y, t) - \bar{u}_0^{\sigma}(x, y, t))^2 \right] dx dy + \right. \\ \left. 0.5 \gamma_{v,0}(\bar{v}^{\sigma}(t) - \bar{v}_0^{\sigma}(t))^2 \right] d\sigma. \quad (4-5)$$

【図14】

$$231: \{(\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$232: \alpha_{g_t}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma, |\nabla \tilde{u}^\sigma|), g_t \in G_t$$

$$233: g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$234: g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$235: \beta_s^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma, (s, \nabla \tilde{u}^\sigma))$$

$$236: (s, \nabla \tilde{u}^\sigma(x, y, t))$$

$$237: (s, \nabla \tilde{u}^\sigma(x, y, t))$$

$$238: (x - \Delta t \cdot \tilde{u}^\sigma(x, y, t), y - \Delta t \cdot \tilde{v}^\sigma(t))$$

$$239: (x + \Delta t \cdot \tilde{u}^\sigma(x, y, t), y + \Delta t \cdot \tilde{v}^\sigma(t))$$

$$240: g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma), g_t \in G_t$$

$$241: (x - \Delta t \cdot \tilde{u}^\sigma(x, y, t), y - \Delta t \cdot \tilde{v}^\sigma(t))$$

$$242: (x + \Delta t \cdot \tilde{u}^\sigma(x, y, t), y + \Delta t \cdot \tilde{v}^\sigma(t))$$

$$243: (\Delta \tilde{u}^\sigma(x, y, t), \Delta \tilde{v}^\sigma(t))$$

$$244: (x - \Delta t \cdot (\tilde{u}^\sigma(x, y, t) + \Delta \tilde{u}^\sigma(x, y, t)), y - \Delta t \cdot (\tilde{v}^\sigma(t) + \Delta \tilde{v}^\sigma(t)))$$

$$245: (x + \Delta t \cdot (\tilde{u}^\sigma(x, y, t) + \Delta \tilde{u}^\sigma(x, y, t)), y + \Delta t \cdot (\tilde{v}^\sigma(t) + \Delta \tilde{v}^\sigma(t)))$$

$$246: (\Delta \tilde{u}^\sigma(x, y, t), \Delta \tilde{v}^\sigma(t))$$

$$247: (x - \Delta t \cdot (\tilde{u}^\sigma(x, y, t) - \Delta \tilde{u}^\sigma(x, y, t)), y - \Delta t \cdot (\tilde{v}^\sigma(t) - \Delta \tilde{v}^\sigma(t)))$$

$$248: (x + \Delta t \cdot (\tilde{u}^\sigma(x, y, t) - \Delta \tilde{u}^\sigma(x, y, t)), y + \Delta t \cdot (\tilde{v}^\sigma(t) - \Delta \tilde{v}^\sigma(t)))$$

$$249: (g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2, g_t \in G_t$$

$$250: (g_t^\sigma(x, y, t, \tilde{u}^\sigma + \Delta \tilde{u}^\sigma, \tilde{v}^\sigma + \Delta \tilde{v}^\sigma))^2, g_t \in G_t$$

【図15】

$$251: (g_t^\sigma(x, y, t, \tilde{u}^\sigma - \Delta \tilde{u}^\sigma, \tilde{v}^\sigma - \Delta \tilde{v}^\sigma))^2, g_t \in G_t$$

$$252:$$

$$(g_t^\sigma(x, y, t, \tilde{u}^\sigma + \Delta \tilde{u}^\sigma, \tilde{v}^\sigma + \Delta \tilde{v}^\sigma))^2 + (g_t^\sigma(x, y, t, \tilde{u}^\sigma - \Delta \tilde{u}^\sigma, \tilde{v}^\sigma - \Delta \tilde{v}^\sigma))^2 - 2(g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2, \quad (4-6)$$

$$253: g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$254: \tilde{u}^\sigma = \tilde{u}^\sigma(x, y, t)$$

$$255: \tilde{v}^\sigma = \tilde{v}^\sigma(t)$$

$$256:$$

$$2(\Delta \tilde{u}^\sigma(x, y, t) g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) + \Delta \tilde{v}^\sigma(t) g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2 \quad (4-7)$$

$$257: g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$258: g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$259: g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)$$

$$260: \tilde{u}^\sigma$$

$$261: \tilde{v}^\sigma$$

$$262: (\Delta \tilde{u}^\sigma(x, y, t), \Delta \tilde{v}^\sigma(t))$$

$$263: (s, \nabla' g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)), g_t \in G_t$$

$$264: (s, \nabla' g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)) = s_x g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) + s_y g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) \quad (4-8)$$

$$265: \{(\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

【図16】

$$266: |\nabla \tilde{u}^{\sigma}(x, y, t)|$$

$$267: (s, \nabla \tilde{u}^{\sigma}(x, y, t)), s \in S$$

$$268: g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$269: |\nabla \tilde{u}^{\sigma}(x, y, t)|$$

$$270: g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$271: (s, \nabla \tilde{u}^{\sigma}(x, y, t))$$

$$272: (s, \nabla' g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma})), g_t \in G_t$$

$$273: (s, \nabla \tilde{u}^{\sigma}(x, y, t)), s \in S$$

$$274: |\nabla \tilde{u}^{\sigma}(x, y, t)|$$

$$275: (s, \nabla' g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma})), g_t \in G_t$$

$$276: \alpha_{g_t}^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma}, |\nabla \tilde{u}^{\sigma}|), g_t \in G_t$$

$$277: g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$278: |\nabla \tilde{u}^{\sigma}(x, y, t)|$$

$$279: g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$280: \beta_{s, \sigma}^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma}, (s, \nabla \tilde{u}^{\sigma})), s \in S$$

$$281: (s, \nabla \tilde{u}^{\sigma}(x, y, t))$$

$$282: (s, \nabla' g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma}))$$

$$283: (s, \nabla \tilde{u}^{\sigma}(x, y, t))$$

$$284: \alpha_{g_t}^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma}, |\nabla \tilde{u}^{\sigma}|) (g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma}))^2, (4-9)$$

$$285: (g_t^{\sigma}(x, y, t, \tilde{u}^{\sigma}, \tilde{v}^{\sigma}))^2$$

【図17】

$$286: (g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2$$

$$287: \beta_s^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, (s, \nabla \bar{u}^\sigma)) (s, \nabla \bar{u}^\sigma(x, y, t))^2 \quad (4-10)$$

$$288: (s, \nabla \bar{u}^\sigma(x, y, t))^2$$

$$289: (s, \nabla \bar{u}^\sigma(x, y, t))^2$$

$$290:$$

$$\alpha_{gt}^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, |\nabla \bar{u}^\sigma|) = (r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma(x, y, t)|^2) \\ (g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2)^{-1}, \quad (4-11)$$

$$291:$$

$$\beta_s^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, (s, \nabla \bar{u}^\sigma)) = \\ (a^2 + (c^2 + b^2 (x, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2) (s, \nabla \bar{u}^\sigma(x, y, t))^2)^{-1}, \quad (4-12)$$

$$292: b^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2$$

$$293: b^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2 = \\ \int_{G_t} b_{g_t}^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2 dg_t, \quad (4-13)$$

$$294: a^2, c^2, b_{g_t}^2, g_t \in G_t$$

$$295: \{(\bar{u}^\sigma(x, y, t, \bar{v}^\sigma(t)) | (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$296: \{(\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) | (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$297: \{(\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) | (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$298: \{(\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) | (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$299: (\bar{u}^\sigma, \bar{v}^\sigma)$$

$$300: \{(\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) | (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

【圖18】

$$301: \{(\tilde{u}_0^\sigma(x, y, t), \tilde{v}_0^\sigma(t)) \mid (x, y) \in \Omega\}$$

$$302:$$

$$(\tilde{u}_0^\sigma(x, y, t), \tilde{v}_0^\sigma(t)) = (\tilde{u}^{\sigma \Delta \sigma}(x, y, t), \tilde{v}^{\sigma \Delta \sigma}(t)), (x, y) \in \Omega$$

(4-14)

$$303: \gamma_{u,0}(\tilde{u}^\sigma(x, y, t) - \tilde{u}_0^\sigma(x, y, t)) = -\gamma_u \frac{\partial}{\partial \sigma} \tilde{u}^\sigma(x, y, t), \quad (4-15)$$

$$304: \gamma_{v,0}(\tilde{v}^\sigma(t) - \tilde{v}_0^\sigma(t)) = -\gamma_v \frac{\partial}{\partial \sigma} \tilde{v}^\sigma(t), \quad (4-16)$$

$$305: \gamma_u = \gamma_{u,0} \Delta \sigma. \quad (4-17)$$

$$306: \gamma_v = \gamma_{v,0} \Delta \sigma. \quad (4-18)$$

$$307: \gamma_{u,0}(\tilde{u}^\sigma(x, y, t) - \tilde{u}_0^\sigma(x, y, t))$$

$$308: \gamma_{v,0}(\tilde{v}^\sigma(t) - \tilde{v}_0^\sigma(t))$$

$$309: -\gamma_u \frac{\partial}{\partial \sigma} \tilde{u}^\sigma(x, y, t)$$

$$310: -\gamma_v \frac{\partial}{\partial \sigma} \tilde{v}^\sigma(t)$$

$$311: \alpha_{g_t}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma, |\nabla \tilde{u}^\sigma|), g_t \in G_t$$

$$312: \beta_{s^\sigma}(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma, (s, \nabla \tilde{u}^\sigma)), s \in S$$

$$313: \{(\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma \in [1, \infty)\}$$

$$314:$$

$$\begin{aligned} & \int_{G_t} \frac{g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)}{r^2 + (p^2 + q^2 |\nabla \tilde{u}^\sigma(x, y, t)|^2) (g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2} dg_t - \\ & \quad \gamma_u \frac{\partial}{\partial \sigma} \tilde{u}^\sigma(x, y, t) - \\ & \int_s (s, \nabla \frac{(s, \nabla \tilde{u}^\sigma(x, y, t))}{a^2 + (c^2 + b^2 (s, \nabla g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2) (s, \nabla \tilde{u}^\sigma(x, y, t))^2}) ds = 0, \\ & \iint_{G_t} \frac{g_{t0}^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma) g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma)}{r^2 + (p^2 + q^2 |\nabla \tilde{u}^\sigma(x, y, t)|^2) (g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2} dg_t dx dy \\ & \quad - \gamma_v \frac{\partial}{\partial \sigma} \tilde{v}^\sigma(t) = 0. \end{aligned} \quad (4-19)$$

【圖19】

$$315: 0.5\gamma_{u,0}(\hat{u}^\sigma(x,y,t) - \bar{u}_0^\sigma(x,y,t))^2$$

$$315-1: 0.5\gamma_{v,0}(\hat{v}^\sigma(t) - \bar{v}_0^\sigma(t))^2$$

$$316: f(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$317: 0.5\gamma_u(\hat{u}^\sigma(x,y,t) - \bar{u}^\sigma(x,y,t) - \frac{\partial}{\partial \sigma} \bar{u}^\sigma(x,y,t))^2$$

$$318: 0.5\gamma_v(\hat{v}^\sigma(t) - \bar{v}^\sigma(t) - \frac{\partial}{\partial \sigma} \bar{v}^\sigma(t))^2$$

$$319: \alpha_{g_c}^\sigma(x,y,t, \bar{u}^\sigma, \bar{v}^\sigma, |\nabla \bar{u}^\sigma|), g_c \in G_c$$

$$320: \beta_s^\sigma(x,y,t, \bar{u}^\sigma, \bar{v}^\sigma, (s, \nabla \bar{u}^\sigma)), s \in S$$

$$321: f(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$322: (\bar{u}, \bar{v}) = \{(\bar{u}^\sigma(x,y,t), \bar{v}^\sigma(t)) \mid (x,y) \in \Omega, \sigma \in [1, \infty)\}$$

$$323: (\bar{u}, \bar{v}) = \{(\bar{u}^\sigma(x,y,t), \bar{v}^\sigma(t)) \mid (x,y) \in \Omega, \sigma \in [1, \infty)\}$$

$$324: h(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$325: (\bar{u}, \bar{v}), (\bar{u}, \bar{v})$$

$$326:$$

$$\begin{aligned} h(\bar{u}, \bar{v}, \bar{u}, \bar{v}) = & \int_1^\infty \left( \int_\Omega \int_{G_c} \frac{\ln(r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma(x,y,t)|^2) (g_c^\sigma(x,y,t, \bar{u}^\sigma, \bar{v}^\sigma))^2)}{2(p^2 + q^2 |\nabla \bar{u}^\sigma(x,y,t)|^2)} dg_c \right. \\ & + \int_S \frac{\ln(a^2 + (c^2 + b^2 (s, \nabla g_c^\sigma(x,y,t, \bar{u}^\sigma, \bar{v}^\sigma))^2) (s, \nabla \bar{u}^\sigma(x,y,t))^2)}{2(c^2 + b^2 (s, \nabla g_c^\sigma(x,y,t, \bar{u}^\sigma, \bar{v}^\sigma))^2)} ds \\ & \left. + 0.5\gamma_u(\hat{u}^\sigma(x,y,t) - \bar{u}^\sigma(x,y,t) - \frac{\partial}{\partial \sigma} \bar{u}^\sigma(x,y,t))^2 \right] dx dy + \\ & 0.5\gamma_v(\hat{v}^\sigma(t) - \bar{v}^\sigma(t) - \frac{\partial}{\partial \sigma} \bar{v}^\sigma(t))^2 d\sigma, \quad (4-20) \end{aligned}$$

$$327: (\bar{u}, \bar{v})$$

$$328: h(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$329: (\bar{u}, \bar{v})$$

$$330: (\bar{u}, \bar{v})$$

【図20】

$$331: (\bar{u}, \bar{v})$$

$$332: (t^n_k)_{k=0}^{K^*}$$

$$333: h(\bar{u}, \bar{v}, \bar{u}, \bar{v})$$

$$334: (\bar{u}, \bar{v})$$

$$335: (\bar{u}, \bar{v}) = ((\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) | (x, y) \in \Omega, \sigma \in [1, \infty))$$

$$336: a_{\sigma}^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, |\nabla \bar{u}^\sigma|), g_t \in G_t$$

$$337: a^2, c^2, b_{g_t}^2, g_t \in G_t$$

$$338: \beta_s^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma, (s, \nabla \bar{u}^\sigma)), s \in S$$

$$339: (\Delta \bar{u}, \Delta \bar{v}) = ((\Delta \bar{u}^\sigma(x, y, t), \Delta \bar{v}^\sigma(t)) | (x, y) \in \Omega, \sigma \in [1, \infty))$$

$$340: (\bar{u}, \bar{v})$$

$$341: h(\bar{u}, \bar{v}, \bar{u} + \Delta \bar{u}, \bar{v} + \Delta \bar{v})$$

$$342: (\Delta \bar{u}, \Delta \bar{v})$$

$$343:$$

$$h(\bar{u}, \bar{v}, \bar{u} + \Delta \bar{u}, \bar{v} + \Delta \bar{v}) - h(\bar{u}, \bar{v}, \bar{u}, \bar{v}) =$$

$$\begin{aligned} & \frac{1}{2} \int_1^\infty \int_\Omega \int_{G_t} \frac{x^2 - (p^2 + q^2 |\nabla \bar{u}^\sigma(x, y, t)|^2) (g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2}{(x^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma(x, y, t)|^2) (g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2)^2} \\ & \quad (g_{t0}^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma) \Delta \bar{u}^\sigma(x, y, t) + \\ & \quad g_{t0}^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma) \Delta \bar{v}^\sigma(t))^2 dg_t \\ & + \int_S \frac{(a^2 - (c^2 + b^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2) (s, \nabla \bar{u}^\sigma(x, y, t))^2)}{(a^2 + (c^2 + b^2 (x, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2) (s, \nabla, \bar{u}^\sigma(x, y, t))^2)^2} \\ & \quad (s, \nabla (\Delta \bar{u}^\sigma(x, y, t)))^2 ds \\ & + \gamma_u (\Delta \bar{u}^\sigma(x, y, t))^2 dx dy + \gamma_v (\Delta \bar{v}^\sigma(t))^2 d\sigma. \quad (4-21) \end{aligned}$$

[圖21]

$$344: g_t^o(x, y, t, \bar{u}^o, \bar{v}^o)$$

$$345:$$

$$\frac{\sqrt{(g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2}}{r^2 + (p^2 + q^2 |\nabla \bar{u}^o(x, y, t)|^2) (g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2} \quad (4-22)$$

$$346: (g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2$$

$$347: \frac{\sqrt{(g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2}}{r^2} \quad (4-23)$$

$$348: g_t^o(x, y, t, \bar{u}^o, \bar{v}^o)$$

$$349: g_t^o(x, y, t, \bar{u}^o, \bar{v}^o)$$

$$350: g_{t0}^o(x, y, t, \bar{u}^o, \bar{v}^o)/r^2$$

$$351: g_{t0}^o(x, y, t, \bar{u}^o, \bar{v}^o) \setminus r^2$$

$$352: (g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2$$

$$352-1: (g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2$$

$$352-2: r^2/p^2 + q^2 |\nabla \bar{u}^o(x, y, t)|^2$$

$$353: g_t^o(x, y, t, \bar{u}^o, \bar{v}^o)$$

$$354: (g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2$$

$$355: r^2/(p^2 + q^2 |\nabla \bar{u}^o(x, y, t)|^2)$$

$$356: g_t^o(x, y, t, \bar{u}^o, \bar{v}^o)$$

$$357: \frac{1}{2r\sqrt{p^2 + q^2 |\nabla \bar{u}^o(x, y, t)|^2}} \quad (4-24)$$

$$358: (g_t^o(x, y, t, \bar{u}^o, \bar{v}^o))^2$$

$$359: r^2/(p^2 + q^2 |\nabla \bar{u}^o(x, y, t)|^2)$$

$$360: |\nabla \bar{u}^o(x, y, t)|$$

【圖22】

$$361: r^2/(p^2 + q^2|\nabla u^\sigma(x, y, t)|^2)$$

$$362: (g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2$$

$$363: (\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t))$$

$$364: (\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t))$$

$$365: g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma)$$

$$366: |\nabla u^\sigma(x, y, t)|$$

$$367: (\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t))$$

$$368: (s, \nabla \bar{u}^\sigma(x, y, t))$$

$$369:$$

$$\frac{\sqrt{(s, \nabla \bar{u}^\sigma(x, y, t))^2}}{a^2 + (c^2 + b^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2) (s, \nabla \bar{u}^\sigma(x, y, t))^2} \quad (4-25)$$

$$370: (s, \nabla \bar{u}^\sigma(x, y, t))^2$$

$$371: \frac{\sqrt{(s, \nabla \bar{u}^\sigma(x, y, t))^2}}{a^2} \quad (4-26)$$

$$372: (s, \nabla \bar{u}^\sigma(x, y, t))$$

$$373: (s, \nabla \bar{u}^\sigma(x, y, t))^2$$

$$374: (s, \nabla \bar{u}^\sigma(x, y, t))^2$$

$$375: a^2/(c^2 + b^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2)$$

$$376: (s, \nabla \bar{u}^\sigma(x, y, t))$$

$$377: (s, \nabla \bar{u}^\sigma(x, y, t))^2$$

$$378: a^2/(c^2 + b^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2)$$

$$379: (s, \nabla \bar{u}^\sigma(x, y, t))$$

$$380: \frac{1}{2a\sqrt{c^2 + b^2 (s, \nabla' g_t^\sigma(x, y, t, \bar{u}^\sigma, \bar{v}^\sigma))^2}} \quad (4-27)$$

【図23】

- 381:  $(s, \nabla \tilde{u}^o(x, y, t))^2$   
 382:  $a^2 / (c^2 + b^2 (s, \nabla' g_t^o(x, y, t, \tilde{u}^o, \tilde{v}^o))^2)$   
 383:  $(s, \nabla' g_t^o(x, y, t, \tilde{u}^o, \tilde{v}^o)), g_t \in G_t$   
 384:  $a^2 / (c^2 + b^2 (s, \nabla' g_t^o(x, y, t, \tilde{u}^o, \tilde{v}^o))^2)$   
 385:  $(s, \nabla \tilde{u}^o(x, y, t))^2$   
 386:  $(s, \nabla \tilde{u}^o(x, y, t))^2$   
 387:  $(\tilde{u}, \tilde{v})$   
 388:  $\alpha_{g_t}^o(x, y, t, \tilde{u}^o, \tilde{v}^o, |\nabla \tilde{u}^o|), g_t \in G_t$   
 389:  $(s, \nabla \tilde{u}^o(x, y, t)), s \in S$   
 390:  $g_t^o(x, y, t, \tilde{u}^o, \tilde{v}^o), g_t \in G_t$   
 391:  $a^2, c^2, b_{g_t}^2, g_t \in G_t$   
 392:  $\beta_{s^o}(x, y, t, \tilde{u}^o, \tilde{v}^o, (s, \nabla \tilde{u}^o)), s \in S$   
 393:  $g_t^o(x, y, t, \tilde{u}^o, \tilde{v}^o), g_t \in G_t$   
 394:  $(s, \nabla \tilde{u}^o(x, y, t)), s \in S$   
 395:  $p^2, q^2, c^2, b_{g_t}^2, g_t \in G_t$   
 396:  $g_t^o(x, y, t, \tilde{u}^o, \tilde{v}^o), g_t \in G_t$   
 397:  $(s, \nabla \tilde{u}^o(x, y, t)), s \in S$   
 398:  $b_{g_t}^2, g_t \in G_t$   
 399:  $((\tilde{u}^o(x, y, t), \tilde{v}^o(t)) | (x, y) \in \Omega, \sigma \in [1, \infty))$   
 400:  $((u^o(x, y, t), v^o(t)) | (x, y) \in \Omega, \sigma \in [1, \infty))$

【図71】

```

arsd[ksp].auu4 = uars + (nsp * 11 + ksp) * nhv;
arsd[ksp].g4sq = uars + (nsp * 11 + ksp) * nhv;
arsd[ksp].auu5 = uars + (nsp * 12 + ksp) * nhv;
arsd[ksp].g5sq = uars + (nsp * 12 + ksp) * nhv;
arsd[ksp].ainv = uars + (nsp * 13 + ksp) * nhv;
arsd[ksp].dusq = uars + (nsp * 13 + ksp) * nhv;
arsd[ksp].cuv1 = uars + (nsp * 14 + ksp) * nhv;
}
wars = uars + 15 * nsp * nhv; j = 0;
for (ksp = 0; ksp < nsp; ksp++)
{
  for (i = 0; i < arsd[ksp].cns; i++)
  {
    arsd[ksp].w11[i] = wars + (nsc * 0 + j) * nhv;
    arsd[ksp].w12[i] = wars + (nsc * 1 + j) * nhv;
    arsd[ksp].w21[i] = wars + (nsc * 2 + j) * nhv;
    arsd[ksp].w22[i] = wars + (nsc * 3 + j) * nhv;
    j++;
  }
}
work = wars + 4 * nsc * nhv;
wrk00 = work; work += nhv;
wrk01 = work; work += nhv;
wrk02 = work; work += nhv;
wrk03 = work; work += nhv;
wrk04 = work; work += nhv;
wrk05 = work; work += nhv;
wrk06 = work; work += nhv;
wrk07 = work; work += nhv;
wrk08 = work; work += nhv;
wrk09 = work; work += nhv;
wrk10 = work; work += nhv;
wrk11 = work; work += nhv;
wrk12 = work; work += nhv;
wrk13 = work; work += nhv;
wrk14 = work; work += nhv;
wrk15 = work; work += nhv;
wrk16 = work; work += nhv;
mars = (long *) work; j = 0;
for (ksp = 0; ksp < nsp; ksp++)
{
  for (i = 0; i < arsd[ksp].cns; i++)
  {
    arsd[ksp].m11[i] = mars + (nsc * 0 + j) * nhv;
    arsd[ksp].m12[i] = mars + (nsc * 1 + j) * nhv;
    arsd[ksp].m21[i] = mars + (nsc * 2 + j) * nhv;
    arsd[ksp].m22[i] = mars + (nsc * 3 + j) * nhv;
    j++;
  }
}
mars += 4 * nsc * nhv;
mv0h0 = mars; mars += nhv;
mvphp = mars; mars += nhv;
mv0hp = mars; mars += nhv;
mvmh0p = mars; mars += nhv;

```

【図24】

$$\begin{aligned}
401: & \int_{G_c} \frac{\bar{g}_{t0}^o \bar{g}_c^o dg_c}{r^2 + (p^2 + q^2 |\nabla \bar{u}^o|^2) (\bar{g}_c^o)^2} - \\
& \int_s (s, \nabla \frac{(s, \nabla \bar{u}^o)}{a^2 + (c^2 + b^2 (s, \nabla \bar{g}_c^o)^2) (s, \nabla \bar{u}^o)^2}) ds \\
& - \gamma_u \frac{\partial}{\partial \sigma} \bar{u}^o = 0, \\
& \iiint_{G_c} \frac{\bar{g}_{t0}^o \bar{g}_c^o dg_c dx dy}{r^2 + (p^2 + q^2 |\nabla \bar{u}^o|^2) (\bar{g}_c^o)^2} - \gamma_v \frac{\partial}{\partial \sigma} \bar{v}^o = 0. \quad (5-1)
\end{aligned}$$

$$402: \bar{g}_1^o, \bar{g}_{t0}^o, \bar{g}_{t0}^o, \bar{g}_t^o$$

$$403: g_t^o, g_{t0}^o, g_{t0}^o, g_t^o$$

$$404: (\bar{u}^o(x, y, t), \bar{v}^o(t))$$

$$405: (u^o(x, y, t), v^o(t))$$

$$406: ((\bar{u}^o(x, y, t), \bar{v}^o(t)) | (x, y) \in \Omega, \sigma \in [1, \infty))$$

$$407: ((\bar{u}^o(x, y, t), \bar{v}^o(t)) | (x, y) \in \Omega, \sigma = \sigma_1, \dots, \sigma_n)$$

$$408: (\bar{u}^o, \bar{v}^o)$$

$$409: \delta(\sigma_i) = \sigma_{i-1}, \quad i=1, \dots, n \quad (5-2)$$

$$410: \gamma_u^o, \gamma_v^o, \sigma = \sigma_1, \dots, \sigma_n$$

$$411: \gamma_u = \gamma_u^o(\delta(\sigma) - \sigma) \quad (5-3)$$

$$412: \gamma_v = \gamma_v^o(\delta(\sigma) - \sigma) \quad (5-4)$$

$$413: -\gamma_u \frac{\partial}{\partial \sigma} \bar{u}^o(x, y, t)$$

$$414: -\gamma_v \frac{\partial}{\partial \sigma} \bar{v}^o(t)$$

$$415: -\gamma_u \frac{\partial}{\partial \sigma} \bar{u}^o(x, y, t) = \gamma_u^o(\bar{u}^o(x, y, t) - \bar{u}^{\delta(\sigma)}(x, y, t)) \quad (5-5)$$

【図25】

$$416: -\gamma_v \frac{\partial}{\partial \sigma} \bar{v}^\sigma(t) = \gamma_v^\sigma (\bar{v}^\sigma(t) - \bar{v}^{\delta(\sigma)}(t)). \quad (5-6)$$

$$417: \{(\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) \mid (x, y) \in \Omega, \sigma = \sigma_1, \dots, \sigma_n\}$$

$$418: \int_{\partial \Omega} \frac{\bar{g}_{\epsilon \alpha} \bar{g}_{\epsilon}^\sigma d g_{\epsilon}}{r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma|^2) (\bar{g}_{\epsilon}^\sigma)^2} - \int_s (s, \nabla \frac{(s, \nabla \bar{u}^\sigma)}{a^2 + (c^2 + b^2 (s, \nabla \bar{g}_{\epsilon}^\sigma)^2) (s, \nabla \bar{u}^\sigma)^2}) ds + \gamma_u^\sigma (\bar{u}^\sigma - \bar{u}^{\delta(\sigma)}) = 0, \\ \iint_{\partial \Omega} \frac{\bar{g}_{\epsilon \alpha} \bar{g}_{\epsilon}^\sigma d g_{\epsilon} dx dy}{r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma|^2) (\bar{g}_{\epsilon}^\sigma)^2} + \gamma_v^\sigma (\bar{v}^\sigma - \bar{v}^{\delta(\sigma)}) = 0. \quad (5-7)$$

$$419: \{(\bar{u}^{\delta(\sigma)}(x, y, t), \bar{v}^{\delta(\sigma)}(t)) \mid (x, y) \in \Omega\}$$

$$420: \{(\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) \mid (x, y) \in \Omega\}$$

$$421: (\bar{u}^\sigma, \bar{v}^\sigma) \equiv \{(\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t)) \mid (x, y) \in \Omega\}$$

$$422: (\bar{u}^\sigma(x, y, t), \bar{v}^\sigma(t))$$

$$423: \mathcal{F}^\sigma(\bar{u}^\sigma, \bar{v}^\sigma) = 0 \quad (5-8)$$

$$424: (\bar{u}^\sigma, \bar{v}^\sigma)$$

$$425: (\bar{u}^{\delta(\sigma)}, \bar{v}^{\delta(\sigma)})$$

$$426: (\bar{u}^{\delta(\sigma)}, \bar{v}^{\delta(\sigma)})$$

$$427: (\bar{u}^\sigma, \bar{v}^\sigma)$$

$$428: (\bar{u}_+^\sigma, \bar{v}_+^\sigma) \equiv \{(\bar{u}_+^\sigma(x, y, t), \bar{v}_+^\sigma(t)) \mid (x, y) \in \Omega\}$$

$$428-1: (\bar{u}_+^\sigma, \bar{v}_+^\sigma) = (\bar{u}^\sigma, \bar{v}^\sigma) + \mathcal{M}(\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma) \quad (5-9)$$

$$428-2: \mathcal{M}$$

$$429: (\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma) \equiv \{(\Delta \bar{u}^\sigma(x, y, t), \Delta \bar{v}^\sigma(t)) \mid (x, y) \in \Omega\}$$

$$429-1: \mathcal{M} \in (0, 1]$$

【図26】

$$430: f^0(\tilde{\omega}) = F^0(\tilde{u}^0 + \tilde{\omega} \Delta \tilde{u}^0, \tilde{v}^0 + \tilde{\omega} \Delta \tilde{v}^0) - F^0(\tilde{u}^0 + \tilde{\omega} \Delta \tilde{u}^0, \tilde{v}^0 + \tilde{\omega} \Delta \tilde{v}^0) \quad (5-10)$$

$$431: (\Delta \tilde{u}^0, \Delta \tilde{v}^0)$$

$$432: (\tilde{u}_*,^0, \tilde{v}_*,^0)$$

$$433: (\tilde{u}^0, \tilde{v}^0)$$

$$434: (\tilde{u}_*,^0, \tilde{v}_*,^0)$$

$$435: (\tilde{u}_*,^0, \tilde{v}_*,^0)$$

$$436: (\tilde{u}^0, \tilde{v}^0)$$

$$437: (\Delta \tilde{u}^0, \Delta \tilde{v}^0)$$

$$438: (\tilde{u}^0, \tilde{v}^0)$$

$$439: F^0$$

$$440: F^0(\tilde{u}^0 + \Delta \tilde{u}^0, \tilde{v}^0 + \Delta \tilde{v}^0) = F^0(\tilde{u}^0, \tilde{v}^0) + J^0(\tilde{u}^0, \tilde{v}^0, \Delta \tilde{u}^0, \Delta \tilde{v}^0) \quad (5-11)$$

$$441: (\Delta \tilde{u}^0, \Delta \tilde{v}^0)$$

$$442: J^0(\tilde{u}^0, \tilde{v}^0, \Delta \tilde{u}^0, \Delta \tilde{v}^0)$$

$$443: F^0(\tilde{u}^0, \tilde{v}^0)$$

$$444: J^0(\tilde{u}^0, \tilde{v}^0, \Delta \tilde{u}^0, \Delta \tilde{v}^0)$$

$$445: J^0(\tilde{u}^0, \tilde{v}^0, \Delta \tilde{u}^0, \Delta \tilde{v}^0) = J^0(\tilde{u}^0, \tilde{v}^0) (\Delta \tilde{u}^0, \Delta \tilde{v}^0) \quad (5-12)$$

$$446: (\Delta \tilde{u}^0, \Delta \tilde{v}^0)$$

$$447: J^0(\tilde{u}^0, \tilde{v}^0) (\Delta \tilde{u}^0, \Delta \tilde{v}^0) = -F^0(\tilde{u}^0, \tilde{v}^0) \quad (5-13)$$

$$448: (\tilde{u}_*,^0, \tilde{v}_*,^0)$$

$$449: \tilde{\omega}$$

$$450: (\tilde{u}_*,^0, \tilde{v}_*,^0)$$

【図27】

$$451: J^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$452: F^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$453: (\Delta \tilde{u}^{\sigma}, \Delta \tilde{v}^{\sigma})$$

$$454: M^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma}) (\Delta \tilde{u}^{\sigma}, \Delta \tilde{v}^{\sigma}) = -F^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma}) \quad (5-14)$$

$$455: (\Delta \tilde{u}^{\sigma}, \Delta \tilde{v}^{\sigma})$$

$$456: M^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma}) (\Delta \tilde{u}^{\sigma}, \Delta \tilde{v}^{\sigma})$$

$$457: J^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma}) (\Delta \tilde{u}^{\sigma}, \Delta \tilde{v}^{\sigma})$$

$$458: F^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$459: F_{\theta}^{\sigma}(\tilde{u}^{\sigma}, \tilde{v}^{\sigma}, \tilde{u}^{\sigma}, \tilde{v}^{\sigma}), \theta \in [0, 1]$$

$$460: (\tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$461: (\tilde{u}^{\sigma}, \tilde{v}^{\sigma})$$

$$462:$$

$$\begin{aligned} & \frac{\int_{\partial \Omega} \tilde{g}_{\varepsilon}^{\sigma} \tilde{g}_{\varepsilon}^{\sigma} d\tilde{g}_{\varepsilon}}{(x^2 + (p^2 + q^2 |\nabla \tilde{u}^{\sigma}|^2) (\tilde{g}_{\varepsilon}^{\sigma})^2)^{1-\theta} (x^2 + (p^2 + q^2 |\nabla \tilde{u}^{\sigma}|^2) (\tilde{g}_{\varepsilon}^{\sigma})^2)^{\theta}} \\ & + \gamma_{\varepsilon}^{\sigma} (\tilde{u}^{\sigma} - \tilde{u}^{\sigma}(\infty)) \\ & - \int_{\Omega} (s, \nabla \frac{\tilde{g}_{\varepsilon}^{\sigma} \tilde{g}_{\varepsilon}^{\sigma} d\tilde{g}_{\varepsilon}}{(a^2 + (c^2 + b^2 (s, \nabla \tilde{g}_{\varepsilon}^{\sigma})^2) (s, \nabla \tilde{u}^{\sigma})^2) (s, \nabla \tilde{u}^{\sigma})^2)^{1-\theta} (a^2 + (c^2 + b^2 (s, \nabla \tilde{g}_{\varepsilon}^{\sigma})^2) (s, \nabla \tilde{u}^{\sigma})^2)^{\theta}} \\ & \frac{\int_{\partial \Omega} \tilde{g}_{\varepsilon}^{\sigma} \tilde{g}_{\varepsilon}^{\sigma} d\tilde{g}_{\varepsilon} dx dy}{(x^2 + (p^2 + q^2 |\nabla \tilde{u}^{\sigma}|^2) (\tilde{g}_{\varepsilon}^{\sigma})^2)^{1-\theta} (x^2 + (p^2 + q^2 |\nabla \tilde{u}^{\sigma}|^2) (\tilde{g}_{\varepsilon}^{\sigma})^2)^{\theta}} \\ & + \gamma_{\varepsilon}^{\sigma} (\tilde{v}^{\sigma} - \tilde{v}^{\sigma}(\infty)), \end{aligned} \quad (5-15)$$

$$463: \tilde{g}_{\varepsilon}^{\sigma}, \tilde{g}_{\varepsilon 0}^{\sigma}, \tilde{g}_{\varepsilon \nu}^{\sigma}, \tilde{g}_{\varepsilon}^{\sigma}$$

$$464: (\tilde{u}^{\sigma}(x, y, t), \tilde{v}^{\sigma}(t))$$

$$465: (u^{\sigma}(x, y, t), v^{\sigma}(t))$$

【図28】

```

466:   $\hat{q}_t^\sigma$ 
467:   $(\hat{u}^\sigma(x, y, t), \hat{v}^\sigma(t))$ 
468:   $(\hat{u}^\sigma(x, y, t), \hat{v}^\sigma(t))$ 
469:   $(\hat{u}^\sigma, \hat{v}^\sigma)$ 
470:   $(\hat{u}^\sigma, \hat{v}^\sigma)$ 
471:   $F_{\theta}^\sigma(\hat{u}^\sigma, \hat{v}^\sigma, \hat{u}^\sigma, \hat{v}^\sigma)$ 
472:   $F^\sigma(\hat{u}^\sigma, \hat{v}^\sigma)$ 
473:   $(\hat{u}^\sigma, \hat{v}^\sigma)$ 
474:   $M_{\theta}^\sigma(\hat{u}^\sigma, \hat{v}^\sigma) (\Delta \hat{u}^\sigma, \Delta \hat{v}^\sigma), \theta \in [0, 1]$ 
475:   $(\Delta \hat{u}^\sigma, \Delta \hat{v}^\sigma)$ 
476:   $M_{\theta}^\sigma(\hat{u}^\sigma, \hat{v}^\sigma) (\Delta \hat{u}^\sigma, \Delta \hat{v}^\sigma)$ 
477:   $F_{\theta}^\sigma(\hat{u}^\sigma, \hat{v}^\sigma, \hat{u}^\sigma, \hat{v}^\sigma)$ 
478:   $(\hat{u}^\sigma, \hat{v}^\sigma)$ 
479:   $(\hat{u}^\sigma, \hat{v}^\sigma)$ 
480:   $(\hat{u}^\sigma, \hat{v}^\sigma)$ 
481:   $(\hat{u}^\sigma, \hat{v}^\sigma)$ 
482:   $M^\sigma(\hat{u}^\sigma, \hat{v}^\sigma) (\Delta \hat{u}^\sigma, \Delta \hat{v}^\sigma)$ 
483:   $M_{\theta}^\sigma(\hat{u}^\sigma, \hat{v}^\sigma) (\Delta \hat{u}^\sigma, \Delta \hat{v}^\sigma), \theta \in [0, 1]$ 
484:   $(\Delta \hat{u}^\sigma, \Delta \hat{v}^\sigma)$ 
485:   $h(\hat{u}^\sigma, \hat{v}^\sigma, \hat{u}^\sigma, \hat{v}^\sigma)$ 

```

【図51】

```

{
    if ((*(argv[k]+3) == 'd') && (*(argv[k]+4) == 't'))
    {
        fldt = argv[k]+5; fldt++;
        if (fldt > 1)
        {
            fprintf(stderr, "Too many fldt files.\n");
            exit(1);
        }
    }
    else
    if ((*(argv[k]+3) == 'm') && (*(argv[k]+4) == 'c'))
    {
        flmc = argv[k]+5; flmc++;
        if (flmc > 1)
        {
            fprintf(stderr, "Too many flmc files.\n");
            exit(1);
        }
    }
    else
    if ((*(argv[k]+3) == 'p') && (*(argv[k]+4) == 'c'))
    {
        flpc = argv[k]+5; flpc++;
        if (flpc > 1)
        {
            fprintf(stderr, "Too many flpc files.\n");
            exit(1);
        }
    }
    else
    {
        fprintf(stderr, "\007The parameter %s", argv[k]);
        fprintf(stderr, " is not recognizable.\n");
        exit(1);
    }
}
else
if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'i'))
{
    if ((*(argv[k]+2) == 'm') && (*(argv[k]+3) == 'c'))
    {
        if (sscanf(argv[k], "-imc%d", &imc) != 1)
        {
            fprintf(stderr, "\007Can't read %s", argv[k]);
            fprintf(stderr, " it must be -imc<long>\n");
            exit(1);
        }
    }
    else
    if ((*(argv[k]+2) == 'p') && (*(argv[k]+3) == 'c'))
    {
        if (sscanf(argv[k], "-ipc%d", &ipc) != 1)
        {
            fprintf(stderr, "\007Can't read %s", argv[k]);
            fprintf(stderr, " it must be -ipc<long>\n");
            exit(1);
        }
    }
    else
    if ((*(argv[k]+2) == 's') && (*(argv[k]+3) == 'c'))

```

【図29】

486:  $(\bar{u}^\sigma, \bar{v}^\sigma)$ 487:  $(\bar{u}^\sigma, \bar{v}^\sigma)$ 488:  $(\bar{u}^\sigma, \bar{v}^\sigma)$ 489:  $(\bar{u}^\sigma, \bar{v}^\sigma)$ 490:  $h(\bar{u}^\sigma, \bar{v}^\sigma, \bar{u}^\sigma + \omega \Delta \bar{u}^\sigma, \bar{v}^\sigma + \omega \Delta \bar{v}^\sigma)$ 491:  $-F^\sigma(\bar{u}^\sigma, \bar{v}^\sigma) {}^2M^\sigma(\bar{u}^\sigma, \bar{v}^\sigma)^{-1} f^\sigma(\bar{u}^\sigma, \bar{v}^\sigma)$  (5-16)492:  $M^\sigma(\bar{u}^\sigma, \bar{v}^\sigma) (\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma)$ 493:  $M^\sigma(\bar{u}^\sigma, \bar{v}^\sigma) (\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma)$ 494:  $J^\sigma(\bar{u}^\sigma, \bar{v}^\sigma) (\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma)$ 495:  $F^\sigma(\bar{u}^\sigma, \bar{v}^\sigma)$ 496:  $M_\sigma(\bar{u}^\sigma, \bar{v}^\sigma) (\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma)$ 497:  $M^\sigma(\bar{u}^\sigma, \bar{v}^\sigma) (\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma)$ 498:  $M^\sigma(\bar{u}^\sigma, \bar{v}^\sigma) (\Delta \bar{u}^\sigma, \Delta \bar{v}^\sigma)$ 

$$\begin{aligned}
 499: & \int_{\partial C} \frac{r^2 \bar{g}_{\varepsilon 0}^\sigma (\bar{g}_{\varepsilon 0}^\sigma \Delta \bar{u}^\sigma + \bar{g}_{\varepsilon 0}^\sigma \Delta \bar{v}^\sigma) d\bar{g}_\varepsilon}{r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma|^2) (\bar{g}_\varepsilon^\sigma)^2} - \\
 & \int_s (s, \nabla \frac{a^2 (s, \nabla (\Delta \bar{u}^\sigma))}{(a^2 + (c^2 + b^2 (s, \nabla \bar{g}_\varepsilon^\sigma)^2) (s, \nabla \bar{u}^\sigma)^2)}) ds \\
 & + \gamma_u^\sigma \Delta \bar{u}^\sigma = \int_{\partial C} \frac{-\bar{g}_{\varepsilon 0}^\sigma \bar{g}_\varepsilon^\sigma d\bar{g}_\varepsilon}{r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma|^2) (\bar{g}_\varepsilon^\sigma)^2} + \\
 & \int_s (s, \nabla \frac{(s, \nabla \bar{u}^\sigma)}{a^2 + (c^2 + b^2 (s, \nabla \bar{g}_\varepsilon^\sigma)^2) (s, \nabla \bar{u}^\sigma)^2}) ds - \\
 & \gamma_u^\sigma (\bar{u}^\sigma - \bar{u}^{\delta(\sigma)}), \\
 & \iiint_{\partial C} \frac{r^2 \bar{g}_{\varepsilon 0}^\sigma (\bar{g}_{\varepsilon 0}^\sigma \Delta \bar{u}^\sigma + \bar{g}_{\varepsilon 0}^\sigma \Delta \bar{v}^\sigma) d\bar{g}_\varepsilon dx dy}{r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma|^2) (\bar{g}_\varepsilon^\sigma)^2} + \gamma_v^\sigma \Delta \bar{v}^\sigma = \\
 & \iiint_{\partial C} \frac{-\bar{g}_{\varepsilon 0}^\sigma \bar{g}_\varepsilon^\sigma d\bar{g}_\varepsilon dx dy}{r^2 + (p^2 + q^2 |\nabla \bar{u}^\sigma|^2) (\bar{g}_\varepsilon^\sigma)^2} - \gamma_v^\sigma (\bar{v}^\sigma - \bar{v}^{\delta(\sigma)}), \quad (5-17)
 \end{aligned}$$

500:  $(\bar{u}(x, y, t), \bar{v}(t))$

【図30】

501:

$$(F * \Phi)(x, y, t) = F(\Phi_{x, y, t}), \Phi_{x, y, t}(\bar{x}, \bar{y}, \bar{t}) = \Phi(x - \bar{x}, y - \bar{y}, t - \bar{t}), \\ (\bar{x}, \bar{y}, \bar{t}) \in R^3, \quad (6-1)$$

$$502: \iiint_{R^3} \chi(x, y, t) dx dy dt = 1$$

$$503: \chi(x, y, t) = \psi(x, y, t) / \iiint_{R^3} \psi(\bar{x}, \bar{y}, \bar{t}) d\bar{x} d\bar{y} d\bar{t}, (x, t, t) \in R^3$$

$$504: (t'_k)_{k=0}^{K'}$$

$$505: (t''_k)_{k=0}^{K''}$$

$$506: t \in (t'_k)_{k=0}^{K'}$$

$$507: \gamma_t = (\Gamma_t * \chi)$$

$$508: R^2(h'_{0,1}, h'_{0,2}) = \{(x', y') \mid (x', y') = i_1 h'_{0,1} + i_2 h'_{0,2}, i_1, i_2 \in \mathbb{Z}\} \\ (6-2)$$

$$509: h'_{0,1}, h'_{0,2}$$

$$510: i_1 h'_{0,1} + i_2 h'_{0,2}$$

$$511: i_1, i_2$$

$$512: R^2(h'_{0,1}, h'_{0,2})$$

$$513: \Gamma_{\xi,0}$$

$$514: \gamma_\xi, \xi \in \Xi$$

$$515: \Gamma_{q,0}$$

【図31】

$$516: \gamma_{\eta}, \eta \in H$$

$$517: \Gamma_{\xi, \sigma}$$

$$518: \gamma_{\xi}$$

$$519: \Gamma_{\xi, \sigma}(\phi) = \sum_{(x, y, t) \in R^2} \sum_{(h'_{\sigma, 1}, h'_{\sigma, 2})} \sum_{x(t'_k)_{k=0}^{\infty}} \gamma_{\xi}(x, y, t) \phi(x, y, t). \quad (6-3)$$

$$520: \Gamma_{\eta, \sigma}$$

$$521: \gamma_{\xi}$$

$$522: \Gamma_{\eta, \sigma}(\phi) = \sum_{(x, y, t) \in R^2} \sum_{(h'_{\sigma, 1}, h'_{\sigma, 2})} \sum_{x(t'_k)_{k=0}^{\infty}} \gamma_{\eta}(x, y, t) \phi(x, y, t) \quad (6-4)$$

$$523: \lambda = (\lambda_{\xi}, \lambda | \xi \in \Xi, \eta \in H)$$

$$524: m_{\eta(\lambda)} = \{m_{\xi, x}, m_{\xi, y}, m_{\xi, t}, m_{\eta, x}, m_{\eta, y}, m_{\eta, t} | \xi \in \Xi, \eta \in H\}$$

$$525: \Gamma_{\sigma, \sigma}$$

$$526: \Gamma_{\sigma, \sigma}$$

$$527:$$

$$\Gamma_{\sigma, \sigma}(\phi) = \sum_{\xi \in \Xi} \lambda_{\xi} \frac{\partial^{m_{\xi, x} + m_{\xi, y} + m_{\xi, t}}}{\partial x^{m_{\xi, x}} \partial y^{m_{\xi, y}} \partial t^{m_{\xi, t}}} \Gamma_{\xi, \sigma}(\phi) + \sum_{\eta \in H} \lambda_{\eta} \frac{\partial^{m_{\eta, x} + m_{\eta, y} + m_{\eta, t}}}{\partial x^{m_{\eta, x}} \partial y^{m_{\eta, y}} \partial t^{m_{\eta, t}}} \Gamma_{\eta, \sigma}(\phi) \quad (6-5)$$

$$528: \Gamma_{\sigma, \lambda}$$

$$529: R^2(h'_{\sigma, 1}, h'_{\sigma, 2}), \sigma = \sigma_0, \sigma_1, \dots, \sigma_n$$

$$530: h'_{\sigma, 1}, h'_{\sigma, 2}$$

【図32】

$$531: R^2(h'_{\sigma_0,1}, h'_{\sigma_0,2}) \subset R^2(h'_{\sigma_1,1}, h'_{\sigma_1,2}) \subset \dots \subset R^2(h'_{\sigma_n,1}, h'_{\sigma_n,2}) \quad (6-6)$$

$$532: h'_{\sigma_0,1}, h'_{\sigma_0,2}$$

$$533: h_{\sigma_{k-1},1}, h_{\sigma_{k-1},2}$$

$$534: h'_{\sigma_{k-1},1} = 0.5(h'_{\sigma_{k-1},1} + h'_{\sigma_{k-1},2}) \quad (6-7)$$

$$535: h'_{\sigma_{k-1},2} = 0.5(h'_{\sigma_{k-1},1} - h'_{\sigma_{k-1},2}) \quad (6-8)$$

$$536: h_{\sigma_{k-1},1}, h_{\sigma_{k-1},2}$$

$$537: t \in \{t^k_k\}_{k=0}^K$$

$$538: g^o(x, y, t)$$

$$539: g_x^o(x, y, t), g_y^o(x, y, t)$$

$$540: \Omega''(h''_{\sigma,1}, h''_{\sigma,2}) \cap R^2(h''_{\sigma,1}, h''_{\sigma,2})$$

$$541: h''_{\sigma,1}, h''_{\sigma,2}$$

$$542: h''_{\sigma,1} = k'h'_{\sigma,1} \quad (6-9)$$

$$543: h''_{\sigma,2} = k'h'_{\sigma,2} \quad (6-10)$$

$$544: \Omega''(h''_{\sigma,1}, h''_{\sigma,2}) \subset R^2$$

$$545:$$

$$\Omega''(h''_{\sigma,1}, h''_{\sigma,2}) = \bigcup_{-1.0 \leq \theta_1 \leq 1.0, -1.0 \leq \theta_2 \leq 1.0} \{(x, y) + \theta_1 h''_{\sigma,1} + \theta_2 h''_{\sigma,2} \mid (x, y) \in \Omega\}$$

(6-11)

$$546: g^o(x, y, t)$$

$$547: g^o(x, y, t)$$

$$548: \Gamma_{g,\sigma}(\psi_{x,y,t}^o)$$

$$549: \Gamma_{g,\sigma}$$

$$550: \psi_{x,y,t}^o$$

【図33】

$$551: g_x^o(x, y, t), g_y^o(x, y, t)$$

$$552: g_x^o(x, y, t), g_y^o(x, y, t)$$

$$553: g^o(x, y, t)$$

$$554: \Gamma_{g, o} \left( \frac{\partial}{\partial x} \psi_{x, y, t}^o, \Gamma_{g, o} \left( \frac{\partial}{\partial y} \psi_{x, y, t}^o \right) \right)$$

$$555: \Gamma_{g, o}$$

$$556: \frac{\partial}{\partial x} \psi_{x, y, t}^o, \frac{\partial}{\partial y} \psi_{x, y, t}^o \in \Phi(R^3).$$

$$557: (x, y) = (i_1 + \theta_1) h''_{o, 1} + (i_2 + \theta_2) h''_{o, 2}. \quad (6-12)$$

$$558: g^o(x, y, t)$$

$$559: g_x^o(x, y, t), g_y^o(x, y, t)$$

$$560:$$

$$g^o(x, y, t) = (1 - \theta_2) \left( (1 - \theta_1) g^o(x_{1,1}, y_{1,1}, t) + \theta_1 g^o(x_{1,2}, y_{1,2}, t) \right) \\ + \theta_2 \left( (1 - \theta_1) g^o(x_{2,1}, y_{2,1}, t) + \theta_1 g^o(x_{2,2}, y_{2,2}, t) \right), \quad (6-13)$$

$$561:$$

$$g_x^o(x, y, t) = \left( (1 - \theta_2) \left( (1 - \theta_1) g_x^o(x_{1,1}, y_{1,1}, t) + \theta_1 g_x^o(x_{1,2}, y_{1,2}, t) \right) \right. \\ \left. + \theta_2 \left( (1 - \theta_1) g_x^o(x_{2,1}, y_{2,1}, t) + \theta_1 g_x^o(x_{2,2}, y_{2,2}, t) \right) \right), \quad (6-14)$$

$$562:$$

$$g_y^o(x, y, t) = (1 - \theta_2) \left( (1 - \theta_1) g_y^o(x_{1,1}, y_{1,1}, t) + \theta_1 g_y^o(x_{1,2}, y_{1,2}, t) \right) \\ + \theta_2 \left( (1 - \theta_1) g_y^o(x_{2,1}, y_{2,1}, t) + \theta_1 g_y^o(x_{2,2}, y_{2,2}, t) \right), \quad (6-15)$$

$$563: g^o(x, y, t)$$

$$564: g_x^o(x, y, t), g_y^o(x, y, t)$$

$$565: \Omega''(h''_{o,1}, h''_{o,2}) \cap R^2(h''_{o,1}, h''_{o,2})$$

$$566: (x_{1,1}, y_{1,1}) = i_1 h''_{o,1} + i_2 h''_{o,2} \quad (6-16)$$

【図34】

$$567: (x_{1,2}, y_{1,2}) = (i_1+1)h''_{o,1} + i_2h''_{o,2} \quad (6-17)$$

$$568: (x_{2,1}, y_{2,1}) = i_1h''_{o,1} + (i_2+1)h''_{o,2} \quad (6-18)$$

$$569: (x_{2,2}, y_{2,2}) = (i_1+1)h''_{o,1} + (i_2+1)h''_{o,2} \quad (6-19)$$

$$570: h_{o,1}, h_{o,2}$$

$$571: h_{o,1} = k''h''_{o,1} \quad (6-20)$$

$$572: h_{o,2} = k''h''_{o,2} \quad (6-21)$$

$$573: \Omega(h_{o,1}, h_{o,2}) = \subset \Omega$$

$$574: \Omega(h_{o,1}, h_{o,2}) = \Omega \cap R^2(h_{o,1}, h_{o,2}). \quad (6-22)$$

$$575: \Omega(h_{o,1}, h_{o,2})$$

$$576: ((\tilde{u}^o(x, y, t), \tilde{v}^o(t)) | (x, y) \in \Omega(h_{o,1}, h_{o,2}))$$

$$577: ((\Delta \tilde{u}^o(x, y, t), \Delta \tilde{v}^o(t)) | (x, y) \in \Omega(h_{o,1}, h_{o,2}))$$

$$577-1: R^2(h_{o,1}, h_{o,2})$$

$$578: \int_s (s, \nabla f(x, y, t, s)) ds = \sum_{s \in S} 0.5 (s, \nabla f(x, y, t, s)) \quad (6-24)$$

$$579: R^2(h_{o,1}, h_{o,2})$$

$$580: (s, \nabla f(x, y, t, s)) = \rho_s(f(x+0.5s_x, y+0.5s_y, t, s) - f(x-0.5s_x, y-0.5s_y, t, s)) \quad (6-25)$$

$$581: \rho_s$$

$$582: s_x, s_y$$

$$583: s = (s_x, s_y)$$

$$584: \sum_{s \in S} 0.5 (s, \nabla f(x, y, t, s)) = \sum_{s \in S} \rho_s f(x+0.5s_x, y+0.5s_y, t, s) \quad (6-26)$$

$$585: p(q_t)$$

【図35】

586:  $g_t \in G_t$ 587:  $\rho(g_t)$ 588:  $g_t \in G_t$ 589:  $R^2(h_{\sigma,1}, h_{\sigma,2})$ 

590:

$$\sum_{g_t \in G_t} \frac{\rho(g_t) \tilde{g}_{t0}^{\sigma} \tilde{g}_t^{\sigma}}{r^2 + (p^2 + q^2 |\nabla \bar{u}^{\sigma}|^2) (\tilde{g}_t^{\sigma})^2} -$$

$$\sum_{s \in S} \frac{\rho_s(s, \nabla \bar{u}^{\sigma})}{a^2 + (c^2 + b^2 (s, \nabla' \tilde{g}_t^{\sigma})^2) (s, \nabla \bar{u}^{\sigma})^2} + \gamma_u^{\sigma} (\bar{u}^{\sigma} - \bar{u}^{\delta(\sigma)}) = 0,$$

$$\sum_{(x,y,g_t) \in \Omega \times G_t} \frac{\rho(g_t) \tilde{g}_{t0}^{\sigma} \tilde{g}_t^{\sigma}}{r^2 + (p^2 + q^2 |\nabla \bar{u}^{\sigma}|^2) (\tilde{g}_t^{\sigma})^2} + \gamma_v^{\sigma} (\bar{v}^{\sigma} - \bar{v}^{\delta(\sigma)}) = 0.$$

(6-27)

591:  $R^2(h_{\sigma,1}, h_{\sigma,2})$ 

592:

$$\sum_{g_t \in G_t} \frac{\rho(g_t) r^2 \tilde{g}_{t0}^{\sigma} (\tilde{g}_{t0}^{\sigma} \Delta \bar{u}^{\sigma} + \tilde{g}_{t0}^{\sigma} \Delta \bar{v}^{\sigma})}{(r^2 + (p^2 + q^2 |\nabla \bar{u}^{\sigma}|^2) (\tilde{g}_t^{\sigma})^2)^2} -$$

$$\sum_{s \in S} \frac{\rho_s a^2 (s, \nabla (\Delta \bar{u}^{\sigma}))}{(a^2 + (c^2 + b^2 (s, \nabla' \tilde{g}_t^{\sigma})^2) (s, \nabla \bar{u}^{\sigma})^2)^2} + \gamma_u^{\sigma} \Delta \bar{u}^{\sigma} =$$

$$\sum_{g_t \in G_t} \frac{-\rho(g_t) \tilde{g}_{t0}^{\sigma} \tilde{g}_t^{\sigma}}{r^2 + (p^2 + q^2 |\nabla \bar{u}^{\sigma}|^2) (\tilde{g}_t^{\sigma})^2} +$$

$$\sum_{s \in S} \frac{\rho_s (s, \nabla \bar{u}^{\sigma})}{a^2 + (c^2 + b^2 (s, \nabla' \tilde{g}_t^{\sigma})^2) (s, \nabla \bar{u}^{\sigma})^2} - \gamma_u^{\sigma} (\bar{u}^{\sigma} - \bar{u}^{\delta(\sigma)}),$$

$$\sum_{(x,y,g_t) \in \Omega \times G_t} \frac{\rho(g_t) r^2 \tilde{g}_{t0}^{\sigma} (\tilde{g}_{t0}^{\sigma} \Delta \bar{u}^{\sigma} + \tilde{g}_{t0}^{\sigma} \Delta \bar{v}^{\sigma})}{(r^2 + (p^2 + q^2 |\nabla \bar{u}^{\sigma}|^2) (\tilde{g}_t^{\sigma})^2)^2} + \gamma_v^{\sigma} \Delta \bar{v}^{\sigma} =$$

$$\sum_{(x,y,g_t) \in \Omega \times G_t} \frac{-\rho(g_t) \tilde{g}_{t0}^{\sigma} \tilde{g}_t^{\sigma}}{r^2 + (p^2 + q^2 |\nabla \bar{u}^{\sigma}|^2) (\tilde{g}_t^{\sigma})^2} - \gamma_v^{\sigma} (\bar{v}^{\sigma} - \bar{v}^{\delta(\sigma)}).$$

(6-28)

593:  $(x+0.5s_x, y+0.5s_y, t)$ 594:  $(x, y) \in R^2(h_{\sigma,1}, h_{\sigma,2}), t \in T, s = (s_x, s_y) \in S$ 595:  $\tilde{g}_t^{\sigma}, \tilde{g}_{t0}^{\sigma}, \tilde{g}_{tv}^{\sigma}$

【図36】

$$\begin{aligned}
 596: \quad \bar{g}_t^o &= g_t^o(x, y, t, \bar{u}^o, \bar{v}^o) = \\
 & (g^o(x + \Delta t^+ \bar{u}^o(x, y, t), y + \Delta t^+ \bar{v}^o(t), \\
 & t + \Delta t^+) - g^o(x - \Delta t^- \bar{u}^o(x, y, t), y - \Delta t^- \bar{v}^o(t), \\
 & t - \Delta t^-)) / (\Delta t^+ + \Delta t^-), \quad (6-29)
 \end{aligned}$$

$$\begin{aligned}
 597: \quad \bar{g}_{ta}^o &= g_{ta}^o(x, y, t, \bar{u}^o, \bar{v}^o) = \\
 & (\Delta t^+ g_x^o(x + \Delta t^+ \bar{u}^o(x, y, t), y + \Delta t^+ \bar{v}^o(t), t + \Delta t^+) \\
 & + \Delta t^- g_x^o(x - \Delta t^- \bar{u}^o(x, y, t), y - \Delta t^- \bar{v}^o(t), t - \Delta t^-)) / \\
 & (\Delta t^+ + \Delta t^-). \quad (6-30)
 \end{aligned}$$

$$\begin{aligned}
 598: \quad \bar{g}_{tb}^o &= g_{tb}^o(x, y, t, \bar{u}^o, \bar{v}^o) = \\
 & (\Delta t^+ g_y^o(x + \Delta t^+ \bar{u}^o(x, y, t), y + \Delta t^+ \bar{v}^o(t), t + \Delta t^+) \\
 & + \Delta t^- g_y^o(x - \Delta t^- \bar{u}^o(x, y, t), y - \Delta t^- \bar{v}^o(t), t - \Delta t^-)) / \\
 & (\Delta t^+ + \Delta t^-). \quad (6-31)
 \end{aligned}$$

$$599: |\nabla \bar{u}|^2$$

$$600:$$

$$|\nabla \bar{u}^o|^2 = |\nabla \bar{u}^o(x, y, t)|^2 = \sum_{s \in S} \rho_s (\bar{u}^o(x + s_x, y + s_y, t) - \bar{u}^o(x, y, t))^2. \quad (6-32)$$

$$601: (s, \nabla \bar{u}^o)$$

$$\begin{aligned}
 602: \quad (s, \nabla \bar{u}^o) &= (s, \nabla \bar{u}^o(x + 0.5s_x, y + 0.5s_y, t)) = \\
 & \rho_s (\bar{u}^o(x + s_x, y + s_y, t) - \bar{u}^o(x, y, t)), \quad (6-33)
 \end{aligned}$$

$$603: (s, \nabla(\Delta \bar{u}^o))$$

$$\begin{aligned}
 604: \quad (s, \nabla(\Delta \bar{u}^o)) &= (s, \nabla(\Delta \bar{u}^o(x + 0.5s_x, y + 0.5s_y, t))) = \\
 & \rho_s (\Delta \bar{u}^o(x + s_x, y + s_y, t) - \Delta \bar{u}^o(x, y, t)), \quad (6-34)
 \end{aligned}$$

$$605: b^2(s, \nabla' \bar{g}_t^o)^2$$

$$606: b^2(s, \nabla' \bar{g}_t^o)^2 = \sum_{s \in S} \rho(g_t) b_{g_t}^2(s, \nabla' \bar{g}_t^o)^2 \quad (6-35)$$

【図37】

$$607: (s, \nabla' \tilde{g}_t^\sigma)^2, g_t \in G_t$$

$$608: \begin{aligned} (s, \nabla' \tilde{g}_t^\sigma)^2 = & (s, \nabla' g_t^\sigma(x+0.5s_x, y+0.5s_y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2 = \\ & (s, \nabla' g_t^\sigma(x, y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2 + (s, \nabla' g_t^\sigma(x+s_x, y+s_y, t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2 \end{aligned}$$

(6-36)

$$609: (x+s_x, y+s_y, t)$$

$$610: (s, \nabla' g_t^\sigma(x', y', t, \tilde{u}^\sigma, \tilde{v}^\sigma))^2$$

$$611: \Omega(h_{\sigma,1}, h_{\sigma,2})$$

$$612: R^2(h_{\sigma,1}, h_{\sigma,2})$$

$$613: (x, y) \in \Omega(h_{\sigma,1}, h_{\sigma,2})$$

$$614: s = (s_x, s_y) \in S$$

$$615: (x+s_x, y+s_y)$$

$$616: \Omega(h_{\sigma,1}, h_{\sigma,2})$$

$$617: (\tilde{u}^\sigma(x+s_x, y+s_y, t), \tilde{v}^\sigma(t))$$

$$618: (\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t))$$

$$619: \Omega(h_{\sigma,1}, h_{\sigma,2})$$

$$620: \{(\tilde{u}^\sigma(x, y, t), \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega(h_{\sigma,1}, h_{\sigma,2})\}$$

$$621: \{(\Delta \tilde{u}^\sigma(x, y, t), \Delta \tilde{v}^\sigma(t)) \mid (x, y) \in \Omega(h_{\sigma,1}, h_{\sigma,2})\}$$

$$622: s = (s_x, s_y) \in S$$

$$623: p_s$$

$$624: p_s = \sqrt{(s_x^2 + s_y^2)} / |S|. \quad (6-37)$$

$$625: S = \{h_{\sigma,1}, -h_{\sigma,1}, h_{\sigma,2}, -h_{\sigma,2}\}, \quad (6-38)$$

【図38】

$$626: S = \{h_{\sigma,1}, -h_{\sigma,1}, h_{\sigma,2}, -h_{\sigma,2}, h_{\sigma,1}+h_{\sigma,2}, -h_{\sigma,1}-h_{\sigma,2}, h_{\sigma,1}-h_{\sigma,2}, -h_{\sigma,1}+h_{\sigma,2}\}. \quad (6-39)$$

$$627: \{(\bar{u}^{\sigma_k}(x, y, t), \bar{v}^{\sigma_k}(t)) \mid (x, y) \in \Omega(h_{\sigma_k,1}, h_{\sigma_k,2})\}$$

$$628: \sigma_k$$

$$629: \{(\bar{u}^{\sigma_{k+1}}(x, y, t), \bar{v}^{\sigma_{k+1}}(t)) \mid (x, y) \in \Omega(h_{\sigma_{k+1},1}, h_{\sigma_{k+1},2})\}$$

$$630: \sigma_{k+1}$$

$$631: (x, y) \in \Omega(h_{\sigma_{k+1},1}, h_{\sigma_{k+1},2})$$

$$632: (x, y) \in \Omega(h_{\sigma_k,1}, h_{\sigma_k,2})$$

$$633: (\bar{u}^{\sigma_{k+1}}(x, y, t), \bar{v}^{\sigma_{k+1}}(t))$$

$$634: (\bar{u}^{\sigma_{k+1}}(x, y, t), \bar{v}^{\sigma_{k+1}}(t)) = (\bar{u}^{\sigma_k}(x, y, t), \bar{v}^{\sigma_k}(t)) \quad (6-40)$$

$$635: (x, y) \in \Omega(h_{\sigma_{k+1},1}, h_{\sigma_{k+1},2})$$

$$636: (x, y) \in \Omega(h_{\sigma_k,1}, h_{\sigma_k,2})$$

$$637: (\bar{u}^{\sigma_{k+1}}(x, y, t), \bar{v}^{\sigma_{k+1}}(t))$$

$$638: \begin{aligned} & (\bar{u}^{\sigma_{k+1}}(x, y, t), \bar{v}^{\sigma_{k+1}}(t)) = \\ & \sum_{i_1, i_2=-1, +1} 0.25 (\bar{u}^{\sigma_k}(x_{i_1, i_2}, y_{i_1, i_2}, t), \bar{v}^{\sigma_k}(t)), \end{aligned} \quad (6-41)$$

$$639: (x_{i_1, i_2}, y_{i_1, i_2}) = (x, y) + i_1 h_{\sigma_{k+1},1} + i_2 h_{\sigma_{k+1},2}. \quad (6-42)$$

$$640: \Omega(h_{\sigma,1}, h_{\sigma,2})$$

【図39】

$$641: M^o(\bar{u}^o, \bar{v}^o) (\Delta \bar{u}^o, \Delta \bar{v}^o)^T = -F^o(\bar{u}^o, \bar{v}^o) \quad (7-1)$$

$$642: (\Delta \bar{u}^o, \Delta \bar{v}^o)^T$$

$$643: (\Delta \bar{u}^o, \Delta \bar{v}^o)$$

$$644: ((\Delta \bar{u}^o(x, y, t), \Delta \bar{v}^o(t)) | (x, y) \in \Omega(h_{o,1}, h_{o,2}))$$

$$645: (\Delta \bar{u}^o(x, y, t) | (x, y) \in \Omega(h_{o,1}, h_{o,2}))$$

$$646: \Delta \bar{v}^o(t)$$

$$647: \Omega(h_{o,1}, h_{o,2})$$

$$648: (x, y) \in \Omega(h_{o,1}, h_{o,2})$$

$$649: (x, y) = i_1 h_{o,1} + i_2 h_{o,2}$$

$$650: (x, y) = i_1 h_{o,1} + i_2 h_{o,2} \in \Omega(h_{o,1}, h_{o,2})$$

$$651: (x', y') = i'_1 h_{o,1} + i'_2 h_{o,2} \in \Omega(h_{o,1}, h_{o,2})$$

$$652: D^o(\bar{u}^o, \bar{v}^o)$$

$$653: M^o(\bar{u}^o, \bar{v}^o)$$

$$654: -B^o(\bar{u}^o, \bar{v}^o)$$

$$655: M^o(\bar{u}^o, \bar{v}^o)$$

$$656: M^o(\bar{u}^o, \bar{v}^o) = D^o(\bar{u}^o, \bar{v}^o) - B^o(\bar{u}^o, \bar{v}^o) \quad (7-2)$$

$$657: (D^o(\bar{u}^o, \bar{v}^o) - B^o(\bar{u}^o, \bar{v}^o)) (\Delta \bar{u}^o, \Delta \bar{v}^o)^T = -F^o(\bar{u}^o, \bar{v}^o). \quad (7-3)$$

$$658: C^o(\bar{u}^o, \bar{v}^o)$$

$$659: C^o(\bar{u}^o, \bar{v}^o) = (D^o(\bar{u}^o, \bar{v}^o))^{1/2}, \quad (7-4)$$

$$660: C^o(\bar{u}^o, \bar{v}^o)^T C^o(\bar{u}^o, \bar{v}^o) = D^o(\bar{u}^o, \bar{v}^o). \quad (7-5)$$

【図40】

$$661: (\Delta u^{\sigma}, \Delta v^{\sigma})$$

$$662: (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})^T = C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma}) (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})^T. \quad (7-6)$$

$$663: (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})$$

$$664: C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})^T$$

$$665: (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})$$

$$666: (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})$$

$$667: (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})$$

$$668: (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})^T = C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})^{-T} B^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma}) C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})^{-1} \\ (\Delta \bar{u}^{\sigma}, \Delta \bar{v}^{\sigma})^T = C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})^{-T} B^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma}). \quad (7-7)$$

$$669: G^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})$$

$$670: G^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma}) = C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})^{-T} B^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma}) C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})^{-1}, \quad (7-8)$$

$$671: h^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})$$

$$672: h^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma}) = C^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma})^{-T} F^{\sigma}(\bar{u}^{\sigma}, \bar{v}^{\sigma}); \quad (7-9)$$

$$673: (\Delta u_0^{\sigma}, \Delta v_0^{\sigma})$$

$$674: (\Delta u_{n+1}^{\sigma}, \Delta v_{n+1}^{\sigma})$$

$$675: (\Delta u_n^{\sigma}, \Delta v_n^{\sigma})$$

【図41】

$$676: (\Delta \bar{u}_{n+1}, \Delta \bar{v}_{n+1})^T = G^o(\bar{u}^o, \bar{v}^o) (\Delta \bar{u}_n, \Delta \bar{v}_n)^T - h^o(\bar{u}^o, \bar{v}^o) \quad (7-10)$$

$$677: (\Delta \bar{u}_n^o, \Delta \bar{v}_n^o)$$

$$678: (\Delta \bar{u}^o, \Delta \bar{v}^o)$$

$$679: (\Delta \bar{u}^o, \Delta \bar{v}^o)^T = C^o(\bar{u}^o, \bar{v}^o)^{-1} (\Delta \bar{u}_n^o, \Delta \bar{v}_n^o)^T \quad (7-11)$$

$$680: w_n = (\Delta \bar{u}^o, \Delta \bar{v}^o)^T, \quad G = G^o(\bar{u}^o, \bar{v}^o), \quad h = h^o(\bar{u}^o, \bar{v}^o) \quad (7-12)$$

$$681: w_{n+1} = G w_n - h. \quad (7-13)$$

$$682: w_{n+1} = \rho_n (\gamma_n (G w_n - h) + (1 - \gamma_n) w_n) + (1 - \rho_n) w_{n-1} \quad (7-14)$$

$$683: \rho_n = \frac{\alpha_{n-1} + \alpha_n \beta_n}{\alpha_{n-1}}, n \geq 1, \rho_0 = 1, \quad \gamma_n = \frac{\alpha_n \alpha_{n-1}}{\alpha_{n-1} + \alpha_n \beta_n}, n \geq 1, \gamma_0 = \alpha_0, \quad (7-15)$$

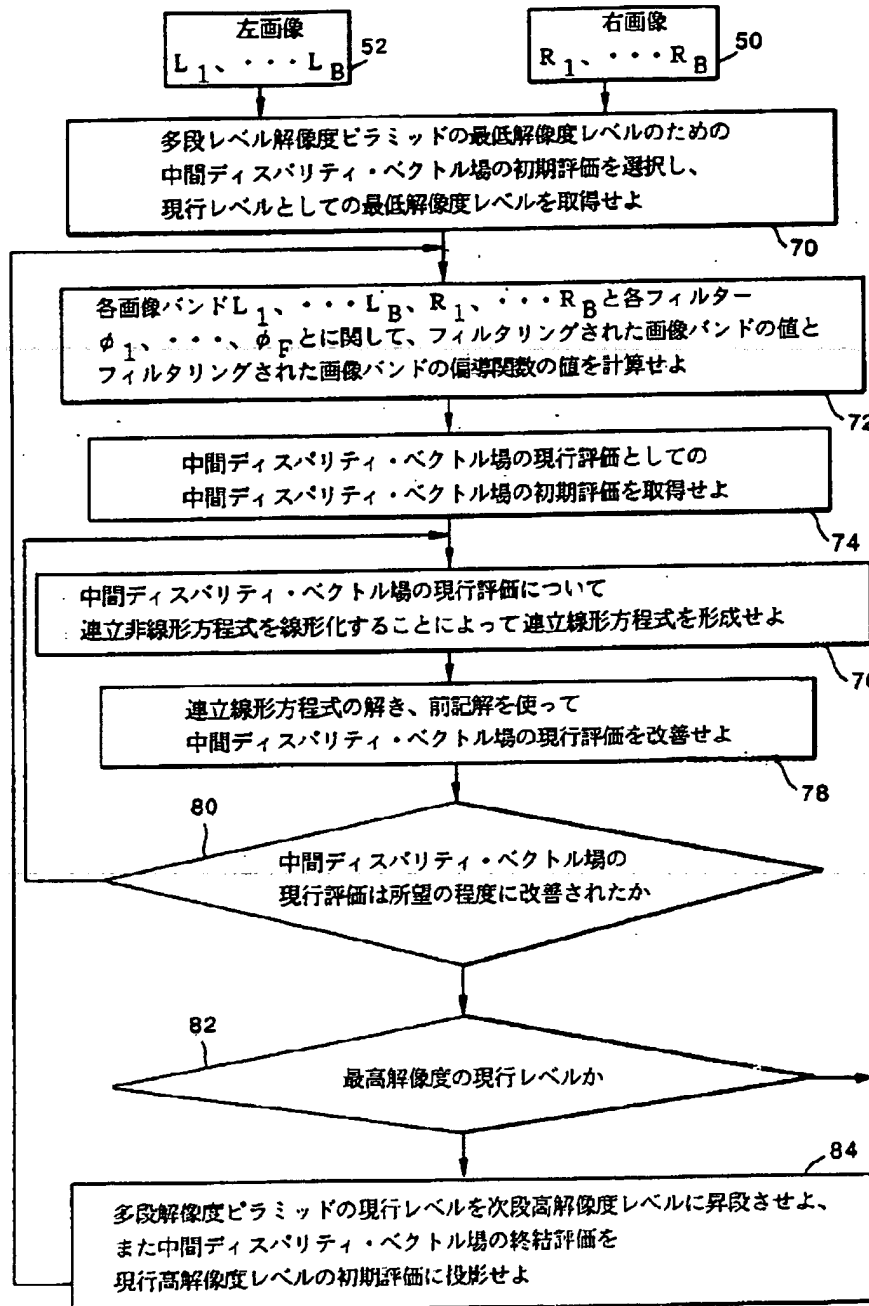
$$684: \alpha_n = \frac{(x_n, x_n)}{(p_n, q_n)}, n \geq 0, \quad \beta_n = \frac{(x_n, x_n)}{(x_{n-1}, x_{n-1})}, n \geq 1, \beta_0 = 0 \quad (7-16)$$

$$685: p_n = x_n + \beta_n p_{n-1}, n \geq 1, p_0 = x_0, \quad q_n = p_n - G p_n, n \geq 0 \quad (7-17)$$

$$686: x_n = G w_n - w_n - h, n \geq 0. \quad (7-18)$$

$$687: w_{n+1} = w_n + \alpha_n p_n, \quad x_{n+1} = x_n - \alpha_n q_n \quad (7-19)$$

【図46】



【図47】

Appendix A--Docket # 66,645

© 1993 Eastman Kodak Company, Rochester, N.Y. 14650-2201

METHOD AND APPARATUS FOR CONSTRUCTING INTERMEDIATE  
IMAGES FOR A DEPTH IMAGE FROM STEREO IMAGES

BY:

Sergei Fogel

© 1993 Eastman Kodak Company

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records but otherwise reserves all copyright rights whatsoever.

Staas & Halsey  
1825 K Street, N.W.  
Washington, D.C. 20006  
202-872-0123

【図108】

```

w12[ktmp] = rw12; temp += rw12; tmpu += tu12 * rw12;
w21[ktmp] = rw21; temp += rw21; tmpu += tu21 * rw21;
w22[ktmp] = rw22; temp += rw22; tmpu += tu22 * rw22;
suma[ktmp] += temp;
smud[ktmp] -= temp;
uvf[ktmp] += tmpu;
}
sumb[0] += kv0b0 * wmn;
smvd[0] -= kv0b0 * wmn;
vvf[0] += kv0b0 * wmn * arsd[ksv].vvf[0];
}
return(0);
)

```

【図48】

```

#define OSIZE 10
#define ASIZE 25
#define BANDS 3
#define BUFEL 4000
#define SMTH 2.0 /* 2.0 */
#define STPD 0.5
#define STEP 0.4
#define SGMb 2.25
#define SGMQ 2.25
#define SGMW 2.25
#define RHG 1.0 /* 0.5 */
#define RHX 3.0 /* 0.5 */
#define DSG 1.0 /* 1.0 */
#define MSG 2.0 /* 1.414213562 */
#include <math.h>
#include <stdio.h>
#include <sys/file.h>
#include "iopackage.h"
#include "cor_corrsv.h"

struct opt_flow {long kxvm[ASIZE], kuvp[ASIZE];
float *ggm[BANDS], *gxm[BANDS], *gym[BANDS], *sxm[BANDS], dhm[ASIZE];
float *xgm[BANDS], *xum[BANDS], *xym[BANDS], *sym[BANDS], dvm[ASIZE];
float *ggp[BANDS], *gxp[BANDS], *gyp[BANDS], *sxp[BANDS], dhp[ASIZE];
float *xgp[BANDS], *xyp[BANDS], *xyp[BANDS], *syp[BANDS], dvp[ASIZE];
float *fgm, *fgp, dtm[ASIZE], duc[ASIZE], dtp[ASIZE];} optf;

struct ars_data {long cns, cnv[ASIZE];
long *m11[ASIZE], *m12[ASIZE], *m21[ASIZE], *m22[ASIZE];
float *w11[ASIZE], *w12[ASIZE], *w21[ASIZE], *w22[ASIZE];
float wmn[ASIZE], wxy[ASIZE], wuv[ASIZE], dlt[ASIZE];
float *auu1, *auu2, *auu3, *auu4, *auu5, *ainv, *bvv1;
float *cuv1, *g2sq, *g3sq, *g4sq, *g5sq, *dusq, *binv;
float *rucv, *pucv, *qucv, *zucv, a, b, bsq, c, csq;
float *rvcv, *pvcv, *qvcv, *zvcv, r, p, psq, q, qsq;
float *uuvf, *u0vf, *udvf, *urvf, *dvrf;
float *vvvf, *v0vf, *vdvf, *vtrf, *dvvf;} arsd[ASIZE];

double sgms, sgds, sgm = 4.0, sgmb, scln, msg = MSG, scl;
double sgmg, sgdg, sgd = 2.0, sgmq, sgmw, dsq = DSG, two = 2.0;

float *hv0h0, *wrk00, *wrk01, *wrk02, *wrk03, *wrk04, *wrk05;
float *vv0h0, *wrk06, *wrk07, *wrk08, *wrk09, *wrk10, *wrk11;
float *wrk12, *wrk13, *wrk14, *wrk15, *wrk16;
long *mv0h0, *mvph0, *mv0hp, *mvmph, *mvmh0, *mvmh1;
long *mv0hm, *mvphm, *mvph0, *m11p, *m22p;
long *m11m, *m12m, *m21m, *m22m, *m12p, *m21p;
float gun = 0.001, gul = 0.01, rhg = RHG, isc = 0.05;
float gvn = 0.001, gvl = 0.01, rhx = RHX, thre;
long kv0h0, kvph0, kv0hp, kvmph, kvmh0, kvmh1, kv0hm, kvphm, kvph0;
long dki = 0, rho = 0, msp = 0, lvs = 0, lev, ksp, nij;
long dkj = 0, tau = 0, nsp = 0, lst = 0, nhv, nxy, bns = 0;

/* The program for estimating the velocity vector fields */
/* from a given time-varying image sequence. */

```

【図49】

```

main(argc,argv)
/* ..... INPUT ..... */
/* nsp = the variable of the integer type, which is equal to */
/* the number of vector fields that are involved in the */
/* computation. */
/* lvs = the variable of the integer type, which is equal to */
/* the number of resolution levels that are used in the */
/* computation. */
/* lst = the variable of the integer type, which is equal to */
/* the resolution level at which the estimation process */
/* stops. */
/* rho = the variable of the integer type, which is equal to */
/* the degree of subsampling of the velocity vector fields */
/* on the finest level of the resolution pyramid relative */
/* to subsampling of the image functions and their partial */
/* derivatives. */
/* tau = the variable of the integer type, which is equal to */
/* the degree of subsampling of the image functions and */
/* their partial derivatives relative to the sampling of */
/* the initial image functions. */
/* ust = the variable of the float type, which is equal to the */
/* x-component of the initial estimate that is used in the */
/* computation. */
/* vst = the variable of the float type, which is equal to the */
/* y-component of the initial estimate that is used in the */
/* computation. */
/* nh = the variable of the integer type, which is equal to */
/* the number of vectors along the horizontal direction */
/* that are involved in the computation. */
/* nv = the variable of the integer type, which is equal to */
/* the number of vectors along the vertical direction */
/* that are involved in the computation. */
/* kx = the variable of the integer type, which is equal to */
/* the number of pixels defining the left and the right */
/* offsets for the image functions and their partial */
/* derivatives; the number kx must be a multiple of the */
/* number two in the power of: lvs in the case when the */
/* number lvs is odd, and lvs + 1 in the case when the */
/* number lvs is even. */
/* ky = the variable of the integer type, which is equal to */
/* the number of pixels defining the lower and the upper */
/* offsets for the image functions and their partial */
/* derivatives; the number ky must be a multiple of the */
/* number two in the power of: lvs in the case when the */
/* number lvs is odd, and lvs + 1 in the case when the */
/* number lvs is even. */
/* nx = the variable of the integer type, which is equal to */
/* the number of vectors along the horizontal direction. */
/* ny = the variable of the integer type, which is equal to */
/* the number of vectors along the vertical direction. */
/* sgm = the variable of the type double specifying the value */
/* of the horizontal component of the sigma factor. */
/* sgd = the variable of the type double specifying the value */
/* of the vertical component of the sigma factor. */
/* msg = the variable of the type double specifying the

```

【図50】

```

/* multiple change factor for the value of the sigma */
/* factor sgm between the resolutions. */
/* dsq = the variable of the type double specifying the */
/* multiple change factor for the value of the sigma */
/* factor sgd between the resolutions. */
/* isc = the variable of the float type specifying the initial */
/* image scale factor. */
/* rhg = the variable of the float type specifying the weight */
/* for the optical flow constraints corresponding to the */
/* image functions. */
/* rhx = the variable of the float type specifying the weight */
/* for the optical flow constraints corresponding to the */
/* x-directional partial derivatives of image functions. */
/* gun = the variable of the float type specifying the weight */
/* for the u-regularization constraints appearing in the */
/* system of nonlinear equations. */
/* gvn = the variable of the float type specifying the weight */
/* for the v-regularization constraints appearing in the */
/* system of nonlinear equations. */
/* gul = the variable of the float type specifying the weight */
/* for the u-regularization constraints appearing in the */
/* system of linear equations. */
/* gvl = the variable of the float type specifying the weight */
/* for the v-regularization constraints appearing in the */
/* system of linear equations. */
/* rep = the variable of the float type defining the criterion */
/* for termination of nonlinear iterations. */
/* fep = the variable of the float type defining the criterion */
/* for termination of linear iterations. */
/* mxn = the variable of the integer type specifying a maximal */
/* number of nonlinear iterations. */
/* mxu = the variable of the integer type specifying a maximal */
/* number of trials allowed in the attempts of updating a */
/* nonlinear iteration. */
/* mxl = the variable of the integer type specifying a maximal */
/* number of linear iterations. */
/* mxs = the variable of the integer type specifying a maximal */
/* number of times the linear system solver can be used on */
/* each nonlinear iteration. */
/* ***** */
int argc;
char *argv[];
{
    unsigned char ;
    char *fldt, *flmc, *flpc;
    double sqrt(), dtmp, stmp;
    long i, n, nx, ki, ni, dd, mh, itst, nh = 0, kcx = 0;
    long j, k, ny, kj, nj, dn, mv, intv, nv = 0, ky = 0;
    long mxn = 0, mxl = 0, imc = 0, jmc = 0, fmc = 0, fdt = 0;
    long mxu = 0, mxs = 0, ipc = 0, jpc = 0, fpc = 0, cldk, bn;
    float avrgm[BANDS], *ggg, *ggx, *ggy, *gsx, tm, ust = 0.0, rep = 0.00001, *fgg, tp;
    float avrgp[BANDS], *gxg, *gxx, *gxy, *gsy, r0, vst = 0.0, fep = 0.00001, temp;
    for ( k = 1; k < argc; k++ )
    {
        if ( (*argv[k]) == '.' && (*argv[k+1]) == 'f' && (*argv[k+2]) == 'l' )

```

【図52】

```

    fprintf(stderr, "it must be -nv<long>\n");
    exit(1);
}
} else
if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'l'))
{
    if ((*(argv[k]+2) == 'v') && (*(argv[k]+3) == 's'))
    {
        if (sscanf(argv[k], "-lvs%d", &lvs) != 1)
        {
            fprintf(stderr, "\007Can't read %s", argv[k]);
            fprintf(stderr, "it must be -lvs<long>\n");
            exit(1);
        }
    } else
    if ((*(argv[k]+2) == 's') && (*(argv[k]+3) == 't'))
    {
        if (sscanf(argv[k], "-lst%d", &lst) != 1)
        {
            fprintf(stderr, "\007Can't read %s", argv[k]);
            fprintf(stderr, "it must be -lst<long>\n");
            exit(1);
        }
    } else
    if ((*(argv[k]) == '-') && (*(argv[k]+1) == 't'))
    {
        if ((*(argv[k]+2) == 'a') && (*(argv[k]+3) == 'u'))
        {
            if (sscanf(argv[k], "-tau%d", &tau) != 1)
            {
                fprintf(stderr, "\007Can't read %s", argv[k]);
                fprintf(stderr, "it must be -tau<long>\n");
                exit(1);
            }
        }
    } else
    if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'u'))
    {
        if ((*(argv[k]+2) == 's') && (*(argv[k]+3) == 't'))
        {
            if (sscanf(argv[k], "-ust%f", &ust) != 1)
            {
                fprintf(stderr, "\007Can't read %s", argv[k]);
                fprintf(stderr, "it must be -ust<float>\n");
                exit(1);
            }
        }
    } else
    if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'v'))
    {
        if ((*(argv[k]+2) == 's') && (*(argv[k]+3) == 't'))
        {
            if (sscanf(argv[k], "-vst%f", &vst) != 1)

```

【図53】

```

    {
        fprintf(stderr, "\007Can't read %s", argv[k]);
        fprintf(stderr, "it must be -vst<float>\n");
        exit(1);
    }
} else
if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'k'))
{
    if ((*(argv[k]+2) == 'x'))
    {
        if (sscanf(argv[k], "-kx%d", &kx) != 1)
        {
            fprintf(stderr, "\007Can't read %s", argv[k]);
            fprintf(stderr, "it must be -kx<long>\n");
            exit(1);
        }
    } else
    if ((*(argv[k]+2) == 'y'))
    {
        if (sscanf(argv[k], "-ky%d", &ky) != 1)
        {
            fprintf(stderr, "\007Can't read %s", argv[k]);
            fprintf(stderr, "it must be -ky<long>\n");
            exit(1);
        }
    }
} else
if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'r'))
{
    if ((*(argv[k]+2) == 'h') && (*(argv[k]+3) == 'o'))
    {
        if (sscanf(argv[k], "-rho%d", &rho) != 1)
        {
            fprintf(stderr, "\007Can't read %s", argv[k]);
            fprintf(stderr, "it must be -rho<long>\n");
            exit(1);
        }
    }
} else
if ((*(argv[k]+2) == 'h') && (*(argv[k]+3) == 'g'))
{
    if (sscanf(argv[k], "-rhg%f", &rhg) != 1)
    {
        fprintf(stderr, "\007Can't read %s", argv[k]);
        fprintf(stderr, "it must be -rhg<float>\n");
        exit(1);
    }
} else
if ((*(argv[k]+2) == 'h') && (*(argv[k]+3) == 'x'))
{
    if (sscanf(argv[k], "-rhx%f", &rhx) != 1)
    {
        fprintf(stderr, "\007Can't read %s", argv[k]);
        fprintf(stderr, "it must be -rhx<float>\n");
        exit(1);
    }
}

```

【図54】

```

    }
} else
if ( (*argv[k]) == '-' && (*argv[k+1]) == 'g' )
{
    if ( (*argv[k+2]) == 'u' && (*argv[k+3]) == 'n' )
    {
        if ( sscanf(argv[k], "-gun%f", &gun) != 1 )
        {
            fprintf(stderr, "007 Can't read %s", argv[k]);
            fprintf(stderr, " it must be -gun<float>\n");
            exit(1);
        }
    } else
    if ( (*argv[k+2]) == 'u' && (*argv[k+3]) == 'l' )
    {
        if ( sscanf(argv[k], "-gul%f", &gul) != 1 )
        {
            fprintf(stderr, "007 Can't read %s", argv[k]);
            fprintf(stderr, " it must be -gul<float>\n");
            exit(1);
        }
    } else
    if ( (*argv[k+2]) == 'v' && (*argv[k+3]) == 'n' )
    {
        if ( sscanf(argv[k], "-gvn%f", &gvn) != 1 )
        {
            fprintf(stderr, "007 Can't read %s", argv[k]);
            fprintf(stderr, " it must be -gvn<float>\n");
            exit(1);
        }
    } else
    if ( (*argv[k+2]) == 'v' && (*argv[k+3]) == 'l' )
    {
        if ( sscanf(argv[k], "-gvl%f", &gvl) != 1 )
        {
            fprintf(stderr, "007 Can't read %s", argv[k]);
            fprintf(stderr, " it must be -gvl<float>\n");
            exit(1);
        }
    }
} else
if ( (*argv[k]) == '-' && (*argv[k+1]) == 'm' )
{
    if ( (*argv[k+2]) == 'x' && (*argv[k+3]) == 'n' )
    {
        if ( sscanf(argv[k], "-mxn%d", &mxn) != 1 )
        {
            fprintf(stderr, "007 Can't read %s", argv[k]);
            fprintf(stderr, " it must be -mxn<long>\n");
            exit(1);
        }
    }
} else
if ( (*argv[k+2]) == 'x' && (*argv[k+3]) == 'u' )
{

```

【図55】

```

    if ( sscanf(argv[k], "-mxu%d", &mxu) != 1 )
    {
        fprintf(stderr, "007 Can't read %s", argv[k]);
        fprintf(stderr, " it must be -mxu<long>\n");
        exit(1);
    }
} else
if ( (*argv[k+2]) == 'x' && (*argv[k+3]) == 'l' )
{
    if ( sscanf(argv[k], "-mxl%d", &mxl) != 1 )
    {
        fprintf(stderr, "007 Can't read %s", argv[k]);
        fprintf(stderr, " it must be -mxl<long>\n");
        exit(1);
    }
} else
if ( (*argv[k+2]) == 'x' && (*argv[k+3]) == 's' )
{
    if ( sscanf(argv[k], "-mxs%d", &mxs) != 1 )
    {
        fprintf(stderr, "007 Can't read %s", argv[k]);
        fprintf(stderr, " it must be -mxs<long>\n");
        exit(1);
    }
}
}
}
if ( ( fmc == 1 ) && ( fpc == 1 ) && ( fdt == 1 ) )
{
    fprintf(stderr, "fldt = %s\n", fldt); fprintf(stdout, "fldt = %s\n", fldt);
    fprintf(stderr, "flmc = %s, lmc = %d, jmc = %d\n", flmc, lmc, jmc);
    fprintf(stdout, "flmc = %s, lmc = %d, jmc = %d\n", flmc, lmc, jmc);
    fprintf(stderr, "flpc = %s, ipc = %d, jpc = %d\n", flpc, ipc, jpc);
    fprintf(stdout, "flpc = %s, ipc = %d, jpc = %d\n", flpc, ipc, jpc);
} else
{
    if ( ( fmc == 0 ) )
    {
        fprintf(stderr, "The file flmc is not specified.\n");
        exit(1);
    }
    if ( ( fpc == 0 ) )
    {
        fprintf(stderr, "The file flpc is not specified.\n");
        exit(1);
    }
}
if ( gul < gun ) gul = gun;
if ( gvl < gvn ) gvl = gvn;
while ( ( nsp <= 0 ) || ( nsp > ASIZE ) )
{
    if ( nsp <= 0 )
    {
        fprintf(stderr, "The parameter nsp either has not been set\n");
        fprintf(stderr, "or has been set to the wrong value; nsp = ");
    }
}

```

【図56】

```

while ( scanf("%d", &nsp) != 1 )
{
    fprintf(stderr, "\007Can't read read input for ");
    fprintf(stderr, "nsp please try again; nsp = \n");
}
if ( nsp > ASIZE )
{
    fprintf(stderr, "The parameter nsp has been set to the value\n");
    fprintf(stderr, "that is greater than the upper limit of the\n");
    fprintf(stderr, "size ASIZE of array of structures arsd; nsp = ");
    while ( scanf("%d", &nsp) != 1 )
    {
        fprintf(stderr, "\007Can't read read input for ");
        fprintf(stderr, "nsp please try again; nsp = \n");
    }
}
while ( lvs < 1 )
{
    fprintf(stderr, "The parameter lvs either has not been set\n");
    fprintf(stderr, "or has been set to the wrong number; lvs = ");
    while ( scanf("%d", &lvs) != 1 )
    {
        fprintf(stderr, "\007Can't read read input for ");
        fprintf(stderr, "lvs please try again; lvs = \n");
    }
}
while ( rho < 1 )
{
    fprintf(stderr, "The parameter rho either has not been set\n");
    fprintf(stderr, "or has been set to the wrong number; rho = ");
    while ( scanf("%d", &rho) != 1 )
    {
        fprintf(stderr, "\007Can't read read input for ");
        fprintf(stderr, "rho please try again; rho = \n");
    }
}
while ( tau < 1 )
{
    fprintf(stderr, "The parameter tau either has not been set\n");
    fprintf(stderr, "or has been set to the wrong number; tau = ");
    while ( scanf("%d", &tau) != 1 )
    {
        fprintf(stderr, "\007Can't read read input for ");
        fprintf(stderr, "tau please try again; tau = \n");
    }
}
while ( nh < 1 )
{
    fprintf(stderr, "The parameter nh either has not been set\n");
    fprintf(stderr, "or has been set to the wrong number; nh = ");
    while ( scanf("%d", &nh) != 1 )
    {
        fprintf(stderr, "\007Can't read read input for ");
    }
}

```

【図60】

```

fprintf(stderr, "\007Can't read %s it ", argv[k]);
fprintf(stderr, "must be: -bsp<float>\n");
exit(1);
}
if ( ( ksp >= 0 ) && ( ksp < nsp ) )
{
    arsd[ksp].b = temp;
} else if ( ksp == nsp )
{
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].b = temp;
    }
    ksp = nsp;
}
} else
if ( (*argv[k]) == '-' && (*(argv[k]+1)) == 'r' )
{
    if ( (*(argv[k]+2)) == 's' && (*(argv[k]+3)) == 'p' )
    {
        if ( sscanf(argv[k], "-rsp%f", &temp) != 1 )
        {
            fprintf(stderr, "\007Can't read %s it ", argv[k]);
            fprintf(stderr, "must be: -rsp<float>\n");
            exit(1);
        }
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        arsd[ksp].r = temp;
    } else if ( ksp == nsp )
    {
        for ( ksp = 0; ksp < nsp; ksp++ )
        {
            arsd[ksp].r = temp;
        }
        ksp = nsp;
    }
}
} else
if ( (*argv[k]) == '-' && (*(argv[k]+1)) == 'p' )
{
    if ( (*(argv[k]+2)) == 's' && (*(argv[k]+3)) == 'p' )
    {
        if ( sscanf(argv[k], "-psp%f", &temp) != 1 )
        {
            fprintf(stderr, "\007Can't read %s it ", argv[k]);
            fprintf(stderr, "must be: -psp<float>\n");
            exit(1);
        }
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        arsd[ksp].p = temp;
    } else if ( ksp == nsp )
    {

```

【図57】

```

    fprintf(stderr, "nh please try again; nh = \n");
}
while (nv < 1)
{
    fprintf(stderr, "The parameter nv either has not been set\n");
    fprintf(stderr, "or has been set to the wrong number; nv = ");
    while (scanf("%d", &nv) != 1)
    {
        fprintf(stderr, "\007Can't read read input for ");
        fprintf(stderr, "nv please try again; nv = \n");
    }
}
for (k = 1; k < argc; k++)
{
    if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'k'))
    {
        if ((*(argv[k]+2) == 's') && (*(argv[k]+3) == 'p'))
        {
            if (sscanf(argv[k], "-ksp%d", &ksp) != 1)
            {
                fprintf(stderr, "\007Can't read %s", argv[k]);
                fprintf(stderr, " it must be -ksp<long>\n");
                exit(1);
            }
            else
            {
                if ((ksp < 0) || (ksp > nsp))
                {
                    fprintf(stderr, "\007ksp = %d is set to wrong number.\n", ksp);
                    exit(1);
                }
            }
        }
        else
        {
            if ((*(argv[k]) == '-') && (*(argv[k]+1) == 'c'))
            {
                if ((*(argv[k]+2) == 'n') && (*(argv[k]+3) == 's'))
                {
                    if (sscanf(argv[k], "-cns%d", &intv) != 1)
                    {
                        fprintf(stderr, "\007Can't read %s", argv[k]);
                        fprintf(stderr, " it must be: -cns<long>\n");
                        exit(1);
                    }
                    else
                    {
                        if ((intv >= 0) && (intv < nsp) && (ksp < nsp))
                        {
                            arsd[ksp].cns = intv;
                        }
                        else
                        {
                            if (ksp == nsp)
                            {
                                fprintf(stderr, "\007ksp = %d is set to wrong number.\n", ksp);
                                exit(1);
                            }
                            else
                            {
                                if ((intv < 0) || (intv >= nsp))
                                {
                                    fprintf(stderr, "\007Can't read %s; cns", argv[k]);
                                    fprintf(stderr, " is set to the wrong number.\n");
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

【図81】

```

    *g5 += *ty * *rxg;
    rgg++; rgx++; tg++;
    rxg++; rnx++; ty++;
}
tg -= kgi2p1; ty -= kgi2p1;
}
tg += rho; ty += rho; mrk = -mrk;
g0 += dd; g1 += dd; g2 += dd;
g3 += dd; g4 += dd; g5 += dd;
}
igg += nx * dd;
}
rump = 0.0; k = 0;
for (j = -ksj; j <= ksj; j++)
{
    tmpys = j * dsc1 / sgdsj;
    dump = -0.5 * tmpys * tmpys;
    rg[k] = exp(dump); rtmp = rtmp + rg[k];
    rty[k] = rg[k] * tmpys / sgdsj;
    k++;
}
rump = sqrt((double)isc) / rtmp; k = 0;
for (j = -ksj; j <= ksj; j++)
{
    rg[k] = rg[k] * rtmp;
    rty[k] = rty[k] * rtmp;
    k++;
}
rump = 0.0; k = 0;
rsg = rggvr; rsx = rgxvr;
for (i = -ksi; i <= ksi; i++)
{
    tmpxs = i * dsc1 / sgmsi;
    dump = -0.5 * tmpxs * tmpxs;
    rsg[k] = exp(dump); rtmp = rtmp + rsg[k];
    rsx[k] = rsg[k] * tmpxs / sgmsi;
    k++;
}
rump = sqrt((double)isc) / rtmp; k = 0;
for (i = -ksi; i <= ksi; i++)
{
    rsg[k] = rsg[k] * rtmp;
    rsx[k] = rsx[k] * rtmp;
    k++;
}
if (dn == dd) mrk = 0; else mrk = 1;
ksit2p1 = 2 * ksi + 1; ksjt2p1 = 2 * ksj + 1;
ifsjdd = ni * (kj - ksj * dd) + ki - ksi * dd; igg = 0;
nsi = rho * (nx - 1) + 2 * ksi + 1;
for (j = 0; j < ny; j += dd)
{
    ifsjdd = ifsjdd;
    tg = tggvr; ty = tgyvr;
    for (i = 0; i < nsi; i += dd)
    {

```

【図58】

```

        exit(1);
    }
}
for ( k = 1; k < argc; k++)
{
    if ( (*argv[k]) == '-' && (*argv[k+1]) == 'k' )
    {
        if ( (*argv[k+2]) == 's' && (*argv[k+3]) == 'p' )
        {
            if ( sscanf(argv[k], "-ksp%d", &ksp) != 1 )
            {
                fprintf(stderr, "007 Can't read %s", argv[k]);
                fprintf(stderr, " it must be -ksp<long>\n");
                exit(1);
            }
            else
            if ( ( ksp < 0 ) || ( ksp > nsp ) )
            {
                fprintf(stderr, "007 ksp = %d is set to wrong number.\n", ksp);
                exit(1);
            }
        }
    }
    else
    if ( (*argv[k]) == '-' && (*argv[k+1]) == 'a' )
    {
        if ( (*argv[k+2]) == 's' && (*argv[k+3]) == 'p' )
        {
            if ( sscanf(argv[k], "-asp%f", &temp) != 1 )
            {
                fprintf(stderr, "007 Can't read %s it ", argv[k]);
                fprintf(stderr, "must be: -asp<float>\n");
                exit(1);
            }
            if ( ( ksp >= 0 ) && ( ksp < nsp ) )
            {
                arsd[ksp].a = temp;
            }
            else if ( ksp == nsp )
            {
                for ( ksp = 0; ksp < nsp; ksp++ )
                {
                    arsd[ksp].a = temp;
                }
                ksp = nsp;
            }
        }
    }
    else
    if ( (*argv[k]) == '-' && (*argv[k+1]) == 'c' )
    {
        if ( (*argv[k+2]) == 's' && (*argv[k+3]) == 'p' )
        {
            if ( sscanf(argv[k], "-csp%f", &temp) != 1 )
            {
                fprintf(stderr, "007 Can't read %s it ", argv[k]);
                fprintf(stderr, "must be: -csp<float>\n");
            }
        }
    }
}

```

【図59】

```

    exit(1);
}
if ( ( ksp >= 0 ) && ( ksp < nsp ) )
{
    arsd[ksp].c = temp;
} else if ( ksp == nsp )
{
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].c = temp;
    }
    ksp = nsp;
} else
if ( (*argv[k]+2) == 'k' ) && (*argv[k]+3) == 'k' )
{
    if ( sscanf(argv[k], "-ckk%d", &ckk) != 1 )
    {
        fprintf(stderr, "007Can't read %s it must be", argv[k]);
        fprintf(stderr, ": it must be -ckk<long>\n");
        exit(1);
    } else
    if ( ksp == nsp )
    {
        fprintf(stderr, "007ksp = %d is set to wrong number.\n", ksp);
        exit(1);
    } else
    if ( ( ckk < 0 ) || ( ckk > arsd[ksp].cns ) )
    {
        fprintf(stderr, "007ckk = %d is set to wrong number.\n", ckk);
        exit(1);
    } else
    if ( (*argv[k]+2) == 'n' ) && (*argv[k]+3) == 'v' )
    {
        if ( sscanf(argv[k], "-cnv%d", &cnv) != 1 )
        {
            fprintf(stderr, "007Can't read %s it must be", argv[k]);
            fprintf(stderr, ": -cnv<long>\n");
            exit(1);
        } else
        if ( ksp == nsp )
        {
            fprintf(stderr, "007ksp = %d is set to wrong number.\n", ksp);
            exit(1);
        }
        arsd[ksp].cnv[ckk] = cnv;
    } else
    if ( (*argv[k]) == '-' ) && (*argv[k]+1) == 'b' )
    {
        if ( (*argv[k]+2) == 's' ) && (*argv[k]+3) == 'p' )
        {
            if ( sscanf(argv[k], "-bsp%f", &temp) != 1 )

```

【図62】

```

    if ( sscanf(argv[k], "-dhp%f", &temp) != 1 )
    {
        fprintf(stderr, "007Can't read %s it must be", argv[k]);
        fprintf(stderr, ": -dhp<float>\n");
        exit(1);
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        optf.dhp[ksp] = temp;
    } else if ( ksp == nsp )
    {
        for ( ksp = 0; ksp < nsp; ksp++ )
        {
            optf.dhp[ksp] = temp;
        }
        ksp = nsp;
    } else
    if ( (*argv[k]+2) == 'v' ) && (*argv[k]+3) == 'm' )
    {
        if ( sscanf(argv[k], "-dvm%f", &temp) != 1 )
        {
            fprintf(stderr, "007Can't read %s it must be", argv[k]);
            fprintf(stderr, ": -dvm<float>\n");
            exit(1);
        }
        if ( ( ksp >= 0 ) && ( ksp < nsp ) )
        {
            optf.dvm[ksp] = temp;
        } else if ( ksp == nsp )
        {
            for ( ksp = 0; ksp < nsp; ksp++ )
            {
                optf.dvm[ksp] = temp;
            }
            ksp = nsp;
        } else
        if ( (*argv[k]+2) == 'v' ) && (*argv[k]+3) == 'p' )
        {
            if ( sscanf(argv[k], "-dvp%f", &temp) != 1 )
            {
                fprintf(stderr, "007Can't read %s it must be", argv[k]);
                fprintf(stderr, ": -dvp<float>\n");
                exit(1);
            }
            if ( ( ksp >= 0 ) && ( ksp < nsp ) )
            {
                optf.dvp[ksp] = temp;
            } else if ( ksp == nsp )
            {
                for ( ksp = 0; ksp < nsp; ksp++ )
                {
                    optf.dvp[ksp] = temp;
                }
            }
        }
    }
}

```

【図61】

```

    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].p = temp;
    }
    ksp = nsp;
}
} else
if ( (*argv[k]) == '-' && (*argv[k+1]) == 'q' )
{
    if ( (*argv[k+2]) == 's' && (*argv[k+3]) == 'p' )
    {
        if ( sscanf(argv[k], "-qsp%f", &temp) != 1 )
        {
            fprintf(stderr, "007Can't read %s it must be", argv[k]);
            fprintf(stderr, "must be: -qsp<float>\n");
            exit(1);
        }
        if ( ( ksp >= 0 ) && ( ksp < nsp ) )
        {
            arsd[ksp].q = temp;
        } else if ( ksp == nsp )
        {
            for ( ksp = 0; ksp < nsp; ksp++ )
            {
                arsd[ksp].q = temp;
            }
            ksp = nsp;
        }
    }
} else
if ( (*argv[k]) == '-' && (*argv[k+1]) == 'd' )
{
    if ( (*argv[k+2]) == 'h' && (*argv[k+3]) == 'm' )
    {
        if ( sscanf(argv[k], "-dhm%f", &temp) != 1 )
        {
            fprintf(stderr, "007Can't read %s it must be", argv[k]);
            fprintf(stderr, "- dhm<float>\n");
            exit(1);
        }
        if ( ( ksp >= 0 ) && ( ksp < nsp ) )
        {
            optf.dhm[ksp] = temp;
        } else if ( ksp == nsp )
        {
            for ( ksp = 0; ksp < nsp; ksp++ )
            {
                optf.dhm[ksp] = temp;
            }
            ksp = nsp;
        }
    }
} else
if ( (*argv[k+2]) == 'h' && (*argv[k+3]) == 'p' )
{

```

【図63】

```

    ksp = nsp;
}
} else
if ( (*argv[k+2]) == 't' && (*argv[k+3]) == 'm' )
{
    if ( sscanf(argv[k], "-dtm%f", &temp) != 1 )
    {
        fprintf(stderr, "007Can't read %s it must be", argv[k]);
        fprintf(stderr, "- dtm<float>\n");
        exit(1);
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        optf.dtm[ksp] = temp;
    } else if ( ksp == nsp )
    {
        for ( ksp = 0; ksp < nsp; ksp++ )
        {
            optf.dtm[ksp] = temp;
        }
        ksp = nsp;
    }
} else
if ( (*argv[k+2]) == 't' && (*argv[k+3]) == 'c' )
{
    if ( sscanf(argv[k], "-dtc%f", &temp) != 1 )
    {
        fprintf(stderr, "007Can't read %s it must be", argv[k]);
        fprintf(stderr, "- dtc<float>\n");
        exit(1);
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        optf.dtc[ksp] = temp;
    } else if ( ksp == nsp )
    {
        for ( ksp = 0; ksp < nsp; ksp++ )
        {
            optf.dtc[ksp] = temp;
        }
        ksp = nsp;
    }
} else
if ( (*argv[k+2]) == 't' && (*argv[k+3]) == 'p' )
{
    if ( sscanf(argv[k], "-dtp%f", &temp) != 1 )
    {
        fprintf(stderr, "007Can't read %s it must be", argv[k]);
        fprintf(stderr, "- dtp<float>\n");
        exit(1);
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        optf.dtp[ksp] = temp;
    } else if ( ksp == nsp )
    {

```

【図64】

```

{
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        optf.dtp[ksp] = temp;
    }
    ksp = nsp;
} else
if ( (*argv[k]+2) == 'l' ) && (*argv[k]+3) == 'r' )
{
    if ( sscanf(argv[k], "-dl:%f", &temp) != 1 )
    {
        fprintf(stderr, "007 Can't read %s it must be", argv[k]);
        fprintf(stderr, ": -dl<float>\n");
        exit(1);
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        arsd[ksp].dlf[ckk] = temp;
    } else if ( ksp == nsp )
    {
        for ( ksp = 0; ksp < nsp; ksp++ )
        {
            arsd[ksp].dlf[ckk] = temp;
        }
        ksp = nsp;
    }
} else
if ( (*argv[k]) == '-' ) && (*argv[k]+1) == 'w' )
{
    if ( (*argv[k]+2) == 'm' ) && (*argv[k]+3) == 'n' )
    {
        if ( sscanf(argv[k], "-wmn%f", &temp) != 1 )
        {
            fprintf(stderr, "007 Can't read %s it must be", argv[k]);
            fprintf(stderr, ": -wmn<float>\n");
            exit(1);
        }
        if ( ( ksp >= 0 ) && ( ksp < nsp ) )
        {
            arsd[ksp].wmn[ckk] = temp;
        } else if ( ksp == nsp )
        {
            for ( ksp = 0; ksp < nsp; ksp++ )
            {
                arsd[ksp].wmn[ckk] = temp;
            }
            ksp = nsp;
        }
    } else
    if ( (*argv[k]+2) == 'x' ) && (*argv[k]+3) == 'y' )
    {
        if ( sscanf(argv[k], "-wxy%f", &temp) != 1 )
        {

```

【図66】

```

{
    fprintf(stderr, "ksp = %d; ", ksp);
    fprintf(stdout, "ksp = %d; ", ksp);
    fprintf(stderr, "a = %6.3f; ", arsd[ksp].a);
    fprintf(stdout, "a = %6.3f; ", arsd[ksp].a);
    fprintf(stderr, "c = %6.3f; ", arsd[ksp].c);
    fprintf(stdout, "c = %6.3f; ", arsd[ksp].c);
    fprintf(stderr, "b = %6.3f; ", arsd[ksp].b);
    fprintf(stdout, "b = %6.3f; ", arsd[ksp].b);
    fprintf(stderr, "r = %6.3f; ", arsd[ksp].r);
    fprintf(stdout, "r = %6.3f; ", arsd[ksp].r);
    fprintf(stderr, "p = %6.3f; ", arsd[ksp].p);
    fprintf(stdout, "p = %6.3f; ", arsd[ksp].p);
    fprintf(stderr, "q = %6.3f; ", arsd[ksp].q);
    fprintf(stdout, "q = %6.3f; ", arsd[ksp].q);
    fprintf(stderr, "cns = %d;\n", arsd[ksp].cns);
    fprintf(stdout, "cns = %d;\n", arsd[ksp].cns);
}
for ( ksp = 0; ksp < nsp; ksp++ )
{
    fprintf(stderr, "ksp = %d; ", ksp);
    fprintf(stdout, "ksp = %d; ", ksp);
    fprintf(stderr, "dhm = %6.3f; ", optf.dhm[ksp]);
    fprintf(stdout, "dhm = %6.3f; ", optf.dhm[ksp]);
    fprintf(stderr, "dvm = %6.3f; ", optf.dvm[ksp]);
    fprintf(stdout, "dvm = %6.3f; ", optf.dvm[ksp]);
    fprintf(stderr, "dtm = %6.3f\n", optf.dtm[ksp]);
    fprintf(stdout, "dtm = %6.3f\n", optf.dtm[ksp]);
    fprintf(stderr, "dtc = %6.3f; ", optf.dtc[ksp]);
    fprintf(stdout, "dtc = %6.3f; ", optf.dtc[ksp]);
    fprintf(stderr, "dhp = %6.3f; ", optf.dhp[ksp]);
    fprintf(stdout, "dhp = %6.3f; ", optf.dhp[ksp]);
    fprintf(stderr, "dvp = %6.3f; ", optf.dvp[ksp]);
    fprintf(stdout, "dvp = %6.3f; ", optf.dvp[ksp]);
    fprintf(stderr, "dtp = %6.3f\n", optf.dtp[ksp]);
    fprintf(stdout, "dtp = %6.3f\n", optf.dtp[ksp]);
}
for ( ksp = 0; ksp < nsp; ksp++ )
{
    for ( ckk = 0; ckk < arsd[ksp].cns; ckk++ )
    {
        fprintf(stderr, "ksp = %d; ckk = %d; ", ksp, ckk);
        fprintf(stdout, "ksp = %d; ckk = %d; ", ksp, ckk);
        fprintf(stderr, "cnv = %d; ", arsd[ksp].cnv[ckk]);
        fprintf(stdout, "cnv = %d; ", arsd[ksp].cnv[ckk]);
        fprintf(stderr, "dlf = %6.3f; ", arsd[ksp].dlf[ckk]);
        fprintf(stdout, "dlf = %6.3f; ", arsd[ksp].dlf[ckk]);
        fprintf(stderr, "wmn = %6.3f; ", arsd[ksp].wmn[ckk]);
        fprintf(stdout, "wmn = %6.3f; ", arsd[ksp].wmn[ckk]);
        fprintf(stderr, "wxy = %6.3f; ", arsd[ksp].wxy[ckk]);
        fprintf(stdout, "wxy = %6.3f; ", arsd[ksp].wxy[ckk]);
        fprintf(stderr, "wuv = %6.3f\n", arsd[ksp].wuv[ckk]);
        fprintf(stdout, "wuv = %6.3f\n", arsd[ksp].wuv[ckk]);
    }
}

```

【図65】

```

    fprintf(stderr, "007Can't read %s it must be", argv[k]);
    fprintf(stderr, ": -wxy<float>\n");
    exit(1);
}
if ( ( ksp >= 0 ) && ( ksp < nsp ) )
{
    arsd[ksp].wxy[ckk] = temp;
} else if ( ksp == nsp )
{
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].wxy[ckk] = temp;
    }
    ksp = nsp;
} else
if ( ( *(argv[k]+2) == 'u' ) && ( *(argv[k]+3) == 'v' ) )
{
    if ( sscanf(argv[k], "-wuv%f", &temp) != 1 )
    {
        fprintf(stderr, "007Can't read %s it must be", argv[k]);
        fprintf(stderr, ": -wuv<float>\n");
        exit(1);
    }
    if ( ( ksp >= 0 ) && ( ksp < nsp ) )
    {
        arsd[ksp].wuv[ckk] = temp;
    } else if ( ksp == nsp )
    {
        for ( ksp = 0; ksp < nsp; ksp++ )
        {
            arsd[ksp].wuv[ckk] = temp;
        }
        ksp = nsp;
    }
}
}

fprintf(stderr, "lvs = %d, rho = %d, ", lvs, rho);
fprintf(stdout, "lvs = %d, rho = %d, ", lvs, rho);
fprintf(stderr, "nh = %d, kx = %d, mxn = %d, mxu = %d\n", nh, kx, mxn, mxu);
fprintf(stdout, "nh = %d, kx = %d, mxn = %d, mxu = %d\n", nh, kx, mxn, mxu);
fprintf(stderr, "nsp = %d, tau = %d, ", nsp, tau);
fprintf(stdout, "nsp = %d, tau = %d, ", nsp, tau);
fprintf(stderr, "nv = %d, ky = %d, mxl = %d, mxs = %d\n", nv, ky, mxl, mxs);
fprintf(stdout, "nv = %d, ky = %d, mxl = %d, mxs = %d\n", nv, ky, mxl, mxs);
fprintf(stderr, "ust = %8.5f, rhg = %8.5f, ", ust, rhg);
fprintf(stdout, "ust = %8.5f, rhg = %8.5f, ", ust, rhg);
fprintf(stderr, "rhx = %8.5f, gun = %8.5f, gul = %8.5f\n", rhx, gun, gul);
fprintf(stdout, "rhx = %8.5f, gun = %8.5f, gul = %8.5f\n", rhx, gun, gul);
fprintf(stderr, "vst = %8.5f, isc = %8.5f, ", vst, isc);
fprintf(stdout, "vst = %8.5f, isc = %8.5f, ", vst, isc);
fprintf(stderr, "gvn = %8.5f, gvl = %8.5f\n", gvn, gvl);
fprintf(stdout, "gvn = %8.5f, gvl = %8.5f\n", gvn, gvl);
for ( ksp = 0; ksp < nsp; ksp++ )

```

【図67】

```

if ( initpr_d(&dd,&dn,&ki,&kj,&nix,&ny,&ni,&nj,&mh,&mv,nh,nv,kx,ky) != 0 ) exit(1);
if ( sdbtm_f(flmc, avrgm, imc - ki, jmc - kj, ni, nj) != 0 ) exit(1);
if ( sdbtm_f(flpc, avrgp, ipc - ki, jpc - kj, ni, nj) != 0 ) exit(1);
if ( istt = initpr_m(ki, dn, kx, ky, nh, nv) != 0 )
{
    fprintf(stderr, "Subroutine initpr_m has failed, istt = %d\n", istt);
    exit(istt);
}
k = 1;
for ( j = 1; j <= nv; j++ )
{
    for ( i = 1; i <= nh; i++ )
    {
        hv0h0[k] = i;
        vv0h0[k] = j;
        k++;
    }
}
for ( ksp = 0; ksp < nsp; ksp++ )
{
    k = 1;
    for ( j = 1; j <= nv; j++ )
    {
        for ( i = 1; i <= nh; i++ )
        {
            arsd[ksp].uuvf[k] = ust;
            k++;
        }
    }
    arsd[ksp].vvvf[0] = vst;
}
for ( lev = lvs; lev > lst; lev-- )
{
    sgdg = sgd; sgds = SMTH * sgd;
    sgmg = sgm; sgms = SMTH * sgm;
    fprintf(stdout, "The level %d: sgmg = %f; sgdg = %f;\n", lev, sgm*scl, sgd*scl);
    fprintf(stderr, "The level %d: sgmg = %f; sgdg = %f;\n", lev, sgm*scl, sgd*scl);
    scln = 2.0 * exp((double)(-1.0));
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].csq = arsd[ksp].c * arsd[ksp].c * scln;
        arsd[ksp].bsq = arsd[ksp].b * arsd[ksp].b * scln * sgmb;
        arsd[ksp].psq = arsd[ksp].p * arsd[ksp].p * scln;
        arsd[ksp].qsq = arsd[ksp].q * arsd[ksp].q * scln * sgmq;
    }
    if ( mxn == 0 ) continue;
    fprintf(stdout, "The derivatives of the images: ");
    fprintf(stderr, "The derivatives of the images: ");
    for ( bn = 0; bn < bns; bn++ )
    {
        if ( rdbtm_f(flmc, avrgm, optf.fgm, imc - ki, jmc - kj, ni, nj, bn) != 0 )
            exit(1);
        fgg = optf.fgm;
        ggg = optf.ggm[bn]; ggx = optf.gxm[bn];
        ggy = optf.gym[bn]; gxy = optf.xgm[bn];
    }
}

```

【図68】

```

gxx = optf.xxm[bn]; gxy = optf.xym[bn];
gsx = optf.sxm[bn]; gsy = optf.sym[bn];
setder(dd,dn,ki,kj,ni,nx,ny,fgg,ggg,ggx,ggg,ggx,gxy,gxx,gsx,gsy);
fprintf(stdout,"fgm[%d], ", bn); fprintf(stderr,"fgm[%d], ", bn);
}
for ( bn = 0; bn < bns; bn++ )
{
    if ( rdbnm_f(flpc, avrgp, optf.fgp, ipc - ki, jpc - kj, ni, nj, bn) != 0 )
        exit(1);
    fgg = optf.fgp;
    ggg = optf.ggp[bn]; gxx = optf.gxp[bn];
    ggy = optf.gyp[bn]; gsx = optf.xgp[bn];
    gxx = optf.xxp[bn]; gxy = optf.xyp[bn];
    gsx = optf.sxp[bn]; gsy = optf.syp[bn];
    setder(dd,dn,ki,kj,ni,nx,ny,fgg,ggg,ggx,ggg,ggx,gxy,gxx,gsx,gsy);
    fprintf(stdout,"fgp[%d], ", bn); fprintf(stderr,"fgp[%d], ", bn);
}
fprintf(stdout,"have been set.\n");
fprintf(stderr,"have been set.\n");
if ( corres(dd,dn,kx,ky,nx,ny,mh,mv,nh,
            nv,rep,fep,mxn,mxu,mxl,mxs) == 1 ) exit(1);
fprintf(stdout,"corres is completed.\n");
fprintf(stderr,"corres is completed.\n");
if ( prctof(&dd, &dn, &mh, &mv, nh) != 0 ) exit(1);
fprintf(stdout,"The level %d is completed.\n", lev);
fprintf(stderr,"The level %d is completed.\n", lev);
sgm = sgm / sqrt(msg); sgmb = sgmb / SGMB;
sgd = sgd / sqrt(dsg); sgmg = sgmg / SGMQ;
scl = scl / sqrt(rwo); sgmw = sgmw / SGMW;
rhx = rhx / (sqrt(msg) * sqrt(two));
}
for ( lev = 1st; lev > 0; lev-- )
{
    if ( prctof(&dd, &dn, &mh, &mv, nh) != 0 ) exit(1);
    fprintf(stdout,"The level %d is completed.\n", lev);
    fprintf(stderr,"The level %d is completed.\n", lev);
}
if ( width(fltd, nh, nv, imc, jmc, ipc, jpc, kx, ky) != 0 )
{
    exit(1);
}
exit(0);

initpr_m(kp, dn, kx, ky, nh, nv)
long kp, dn, kx, ky, nh, nv;
{
    float *gopt, *wars, *uars, *vars, *work, *uuvf, *vvvf;
    long *mars, ktmp, ksp, nsc, bn, i, j;
    ktmp = kx / dn; ktmp = ktmp * dn;
    if ( bns == 0 ) return(1);
    if ( kx != ktmp )
    {
        fprintf(stderr,"The kx should be divisible by %d.\n", dn);
        return(1);
    }
}

```

【図85】

```

for( i = 1; i < *mh; i++ )
{
    k12 = k - ddnh; k11 = k - *dd;
    k21 = k + ddnh; k22 = k + *dd;
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].uuvf[k] = (arsd[ksp].uuvf[k11]
                               + arsd[ksp].uuvf[k12]
                               + arsd[ksp].uuvf[k21]
                               + arsd[ksp].uuvf[k22]) * 0.25;
    }
    k += *dn;
}
k = k1;
for( i = 1; i < *mh; i++ )
{
    k11 = k - *dd; k22 = k + *dd;
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].uuvf[k] = (arsd[ksp].uuvf[k11]
                               + arsd[ksp].uuvf[k22]) * 0.5;
    }
    k += *dn;
}
k1 = *dd * nh + 1;
for ( j = 1; j < *mv; j++ )
{
    k = k1; k1 += ddnh;
    k12 = k - ddnh; k21 = k + ddnh;
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].uuvf[k] = (arsd[ksp].uuvf[k12]
                               + arsd[ksp].uuvf[k21]) * 0.5;
    }
    k += *dn;
}
for( i = 2; i < *mh; i++ )
{
    k12 = k - ddnh; k11 = k - *dd;
    k21 = k + ddnh; k22 = k + *dd;
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        arsd[ksp].uuvf[k] = (arsd[ksp].uuvf[k11]
                               + arsd[ksp].uuvf[k12]
                               + arsd[ksp].uuvf[k21]
                               + arsd[ksp].uuvf[k22]) * 0.25;
    }
    k += *dn;
}
k12 = k - ddnh; k21 = k + ddnh;
for ( ksp = 0; ksp < nsp; ksp++ )
{
    arsd[ksp].uuvf[k] = (arsd[ksp].uuvf[k12]
                          + arsd[ksp].uuvf[k21]) * 0.5;
}
}

```

【図69】

```

    )
    ktmp = ky / dn; ktmp = ktmp * dn;
    if ( ky != ktmp )
    {
        fprintf(stderr, "The ky should be divisible by %d.\n", dn);
        return(1);
    }
    ktmp = (nh - 1) / dn;
    ktmp = ktmp * dn + 1;
    if ( nh != ktmp )
    {
        fprintf(stderr, "The nh - 1 should be divisible by %d.\n", dn);
        return(1);
    }
    ktmp = (nv - 1) / dn;
    ktmp = ktmp * dn + 1;
    if ( nv != ktmp )
    {
        fprintf(stderr, "The nv - 1 should be divisible by %d.\n", dn);
        return(1);
    }
    gopt = (float *) malloc((unsigned)(16 * nxy * bns * sizeof(float)));
    if ( !gopt )
    {
        fprintf(stderr, "memory requested for array gopt failed, msp = %d\n", msp);
        return(1);
    }
    else msp += 16 * nxy * bns * sizeof(float);
    fprintf(stderr, "gopt: msp = %d; ", msp);
    for ( bn = 0; bn < bns; bn++ )
    {
        optf.ggm[bn] = gopt + (bns * 0 + bn) * nxy;
        optf.gxm[bn] = gopt + (bns * 1 + bn) * nxy;
        optf.gym[bn] = gopt + (bns * 2 + bn) * nxy;
        optf.sxm[bn] = gopt + (bns * 3 + bn) * nxy;
        optf.xgm[bn] = gopt + (bns * 4 + bn) * nxy;
        optf.xxm[bn] = gopt + (bns * 5 + bn) * nxy;
        optf.xym[bn] = gopt + (bns * 6 + bn) * nxy;
        optf.sym[bn] = gopt + (bns * 7 + bn) * nxy;
        optf.ggp[bn] = gopt + (bns * 8 + bn) * nxy;
        optf.gxp[bn] = gopt + (bns * 9 + bn) * nxy;
        optf.gyp[bn] = gopt + (bns * 10 + bn) * nxy;
        optf.sxp[bn] = gopt + (bns * 11 + bn) * nxy;
        optf.xgp[bn] = gopt + (bns * 12 + bn) * nxy;
        optf.xyp[bn] = gopt + (bns * 13 + bn) * nxy;
        optf.xyp[bn] = gopt + (bns * 14 + bn) * nxy;
        optf.syp[bn] = gopt + (bns * 15 + bn) * nxy;
    }
    hv0h0 = (float *) malloc((unsigned)(nhv * sizeof(float)));
    if ( !hv0h0 )
    {
        fprintf(stderr, "memory requested for array hv0h0 failed, msp = %d\n", msp);
        return(1);
    }
    else msp += nhv * sizeof(float);
    vv0h0 = (float *) malloc((unsigned)(nhv * sizeof(float)));
    if ( !vv0h0 )

```

【図70】

```

{
    fprintf(stderr, "memory requested for array vv0h0 failed, msp = %d\n", msp);
    return(1);
} else msp += nhv * sizeof(float);
uuvf = (float *) malloc((unsigned)(nsp * nhv * sizeof(float)));
if (!uuvf)
{
    fprintf(stderr, "memory requested for array uuvf failed, msp = %d\n", msp);
    return(1);
} else msp += nsp * nhv * sizeof(float);
for (ksp = 0; ksp < nsp; ksp++)
{
    arsd[ksp].uuvf = uuvf + ksp * nhv;
}
vvvf = (float *) malloc((unsigned)(nsp * sizeof(float)));
if (!vvvf)
{
    fprintf(stderr, "memory requested for array vvvf failed, msp = %d\n", msp);
    return(1);
} else msp += nsp * sizeof(float);
for (ksp = 0; ksp < nsp; ksp++)
{
    arsd[ksp].vvvf = vvvf + ksp;
}
nsc = 0;
for (ksp = 0; ksp < nsp; ksp++)
{
    nsc = nsc + arsd[ksp].cns;
}
ktmp = (15 * nsp + 4 * nsc + 17) * nhv * sizeof(float)
        + (4 * nsc + 17) * nhv * sizeof(long);
if (ktmp < nij * sizeof(float)) ktmp = nij * sizeof(float);
uars = (float *) malloc((unsigned)ktmp);
if (!uars)
{
    fprintf(stderr, "memory requested for array uars failed, msp = %d\n", msp);
    return(1);
} else msp += ktmp;
fprintf(stderr, "uars: msp = %d; ", msp);
optf.fgm = uars; optf.fgp = uars;
for (ksp = 0; ksp < nsp; ksp++)
{
    arsd[ksp].u0vf = uars + (nsp * 0 + ksp) * nhv;
    arsd[ksp].u1vf = uars + (nsp * 1 + ksp) * nhv;
    arsd[ksp].u2vf = uars + (nsp * 2 + ksp) * nhv;
    arsd[ksp].u3vf = uars + (nsp * 3 + ksp) * nhv;
    arsd[ksp].u4vf = uars + (nsp * 4 + ksp) * nhv;
    arsd[ksp].u5vf = uars + (nsp * 5 + ksp) * nhv;
    arsd[ksp].u6vf = uars + (nsp * 6 + ksp) * nhv;
    arsd[ksp].u7vf = uars + (nsp * 7 + ksp) * nhv;
    arsd[ksp].u8vf = uars + (nsp * 8 + ksp) * nhv;
    arsd[ksp].u9vf = uars + (nsp * 9 + ksp) * nhv;
    arsd[ksp].g2sq = uars + (nsp * 9 + ksp) * nhv;
    arsd[ksp].auu3 = uars + (nsp * 10 + ksp) * nhv;
    arsd[ksp].g3sq = uars + (nsp * 10 + ksp) * nhv;
}

```

【図72】

```

mvmh0 = mars; mars += nhv;
mvmhm = mars; mars += nhv;
mv0hm = mars; mars += nhv;
mvphm = mars; mars += nhv;
mvph0 = mars; mars += nhv;
m11m = mars; mars += nhv;
m12m = mars; mars += nhv;
m21m = mars; mars += nhv;
m22m = mars; mars += nhv;
m11p = mars; mars += nhv;
m12p = mars; mars += nhv;
m21p = mars; mars += nhv;
m22p = mars;
vars = (float *) malloc((unsigned)(10 * nsp * sizeof(float)));
if ( !vars )
{
    fprintf(stderr, "memory requested for array vars failed, msp = %d\n", msp);
    return(1);
} else msp += 10 * nsp * sizeof(float);
for ( ksp = 0; ksp < nsp; ksp++ )
{
    arsd[ksp].v0vf = vars + nsp * 0 + ksp;
    arsd[ksp].vdvf = vars + nsp * 1 + ksp;
    arsd[ksp].vvvf = vars + nsp * 2 + ksp;
    arsd[ksp].dvvf = vars + nsp * 3 + ksp;
    arsd[ksp].rvcv = vars + nsp * 4 + ksp;
    arsd[ksp].pvcv = vars + nsp * 5 + ksp;
    arsd[ksp].qvcv = vars + nsp * 6 + ksp;
    arsd[ksp].zvcv = vars + nsp * 7 + ksp;
    arsd[ksp].bvvl = vars + nsp * 8 + ksp;
    arsd[ksp].binv = vars + nsp * 9 + ksp;
}
fprintf(stderr, "msp = %d; nij = %d; nhv = %d; nxy = %d\n", msp, nij, nhv, nxy);
return(0);
}

sdbtm_f(filein, averge, kki, kkj, ni, nj)
char *filein;
float *averge;
long kki, kkj, ni, nj;
{
    unsigned char buffer[BUFEL]; float pxltln;
    long pixels, lines, bands, form, imgin;
    long apixel, aline, aband, arrtyp;
    arrtyp = IDBYTE; imgin = 0;
    if ( opening(&imgin, filein, 0, FALSE) != SYSNRM ) return(1);
    getdef(&imgin, &pixels, &lines, &bands, &form);
    pxltln = pixels * lines;
    if ( kki + dki < 0 ) dki = -kki;
    if ( dki < kki + ni - pixels ) dki = kki + ni - pixels;
    if ( dkj + kkj < 0 ) dkj = -kkj;
    if ( dkj < kkj + nj - lines ) dkj = kkj + nj - lines;
    if ( form != arrtyp )
    {
        fprintf(stderr, "The input is the wrong data type\n");
    }
}

```

【図73】

```

    return(1);
}
for ( aband = 0; aband < bands; aband++ )
{
    averge[aband] = 0.0;
}
for ( aline = 0; aline < lines; aline++ )
{
    for ( aband = 0; aband < bands; aband++ )
    {
        rdline(&imgin, aline, aband, buffer, BUFEL, arryp);
        for ( apixel = 0; apixel < pixels; apixel++ )
        {
            averge[aband] = averge[aband] + buffer[apixel];
        }
    }
}
for ( aband = 0; aband < bands; aband++ )
{
    averge[aband] = averge[aband] / pxldn;
}
clsimg(&imgin);
if ( bns == 0 ) bns = bands;
else if ( bns != bands ) return(1);
return(0);
}

rdbtim_ff(filein, averge, fg, kki, kkj, ni, nj, bn)
char *filein; float *averge, *fg;
long kki, kkj, ni, nj, bn;
{
    unsigned char buffer[BUFEL];
    long nimd, pixels, lines, bands, line, band, imgin;
    long njmd, dpixel, dline, pixel, form, arryp, k;
    nimd = ni - dki;
    njmd = nj - dkj;
    arryp = IDBYTE; imgin = 0;
    if ( opning(&imgin, filein, 0, FALSE) != SYSNRM ) return(1);
    getdef(&imgin, &pixels, &lines, &bands, &form);
    if ( form != arryp )
    {
        fprintf(stderr, "The input is the wrong data type\n");
        return(1);
    }
    if ( bands != bns )
    {
        fprintf(stderr, "The input is the wrong bands type\n");
        return(1);
    }
    if ( bands > BANDS )
    {
        fprintf(stderr, "The input has too many bands\n");
        return(1);
    }
    if ( pixels > BUFEL )

```

【図84】

```

        fprintf(stderr, "(u,v) = (%6.4f,%6.4f)\n", tmpuu, tmpvv);
        fprintf(stdout, "(u,v) = (%6.4f,%6.4f)\n", tmpuu, tmpvv);
    }
    return(0);
}

prctof(dd, dn, mh, mv, nh)
long *dd, *dn, *mh, *mv, nh;
{
    register long k, k11, k12, k21, k22;
    long nhp1, k1, i, j, dntnh, ddtmh, ddtmhp1;
    if ( (*dn == 1) && (*dd == 1) ) return(0);
    nhp1 = nh + 1; dntnh = *dn * nh;
    if ( *dn == *dd )
    {
        *dd = *dn * 0.5;
        ddtmhp1 = *dd * nhp1;
        k1 = *dd * nh + *dd + 1;
        for ( j = 1; j < *mv; j++ )
        {
            k = k1; k1 += dntnh;
            for ( i = 1; i < *mh; i++ )
            {
                k11 = k - ddtmhp1; k12 = k11 + *dn;
                k22 = k + ddtmhp1; k21 = k22 - *dn;
                for ( ksp = 0; ksp < nsp; ksp++ )
                {
                    arsd[ksp].uuvf[k] = (arsd[ksp].uuvf[k11]
                        + arsd[ksp].uuvf[k12]
                        + arsd[ksp].uuvf[k21]
                        + arsd[ksp].uuvf[k22]) * 0.25;
                }
            }
            k += *dn;
        }
    }
    return(0);
}

if ( *dn == (*dd + *dd) )
{
    ddtmh = *dd * nh;
    k1 = *dd + 1;
    k = k1; k1 += dntnh;
    for ( i = 1; i < *mh; i++ )
    {
        k11 = k - *dd; k22 = k + *dd;
        for ( ksp = 0; ksp < nsp; ksp++ )
        {
            arsd[ksp].uuvf[k] = (arsd[ksp].uuvf[k11]
                + arsd[ksp].uuvf[k22]) * 0.5;
        }
        k += *dn;
    }
    for ( j = 2; j < *mv; j++ )
    {
        k = k1; k1 += dntnh;

```

【図74】

```

{
    fprintf(stderr, "The buffer is not large enough\n");
    return(1);
}
if ( kki + dki < 0 )
{
    fprintf(stderr, "The dki = %d is not large enough, ", dki);
    fprintf(stderr, "the value of dki should be = %d\n", -kki);
    return(1);
}
if ( pixels - kki - nimd < 0 )
{
    fprintf(stderr, "The dki = %d is not large enough, the ", dki);
    fprintf(stderr, "value of dki should be = %d\n", kki + ni - pixels);
    return(1);
}
if ( kkj + dkj < 0 )
{
    fprintf(stderr, "The dkj = %d is not large enough, ", dkj);
    fprintf(stderr, "the value of dkj should be = %d\n", -kkj);
    return(1);
}
if ( lines - kkj - njmd < 0 )
{
    fprintf(stderr, "The dkj = %d is not large enough, the ", dkj);
    fprintf(stderr, "value of dkj should be = %d\n", kkj + nj - lines);
    return(1);
}
for ( line = kkj + dkj; line < kkj + njmd; line++ )
{
    for ( band = 0; band < bands; band++ )
    {
        rdline(&imgin, line, band, buffer, BUFEL, arctyp);
        if ( band == bn )
        {
            if ( line == kkj + dkj ) k = 0;
            for ( pixel = kki + dki; pixel < kki + nimd; pixel++ )
            {
                fg[k] = buffer[pixel] - averge[band]; k++;
                if ( dki > 0 )
                {
                    if ( ( pixel == kki + dki ) || ( pixel == kki + nimd - 1 ) )
                    {
                        for ( dpixel = 0; dpixel < dki; dpixel++ )
                        {
                            fg[k] = fg[k - 1]; k++;
                        }
                    }
                }
            }
        }
    }
}
if ( dkj > 0 )
{
    if ( ( line == kkj + dkj ) || ( line == kkj + njmd - 1 ) )
    {
        for ( dline = 0; dline < dkj; dline++ )

```

【図86】

```

}
    *dn = *dd;
    *mh = 2 * (*mh - 1) + 1;
    *mv = 2 * (*mv - 1) + 1;
    return(0);
}
return(1);
}

widthf(fldt, nh, nv, kim, kjm, kip, kjp, kx, ky)
char *fldt;
long nh, nv, kim, kjm, kip, kjp, kx, ky;
{
    float temp, buffer[BUFEL];
    int wrtn; static int fd;
    long k, i, ix, jy, bn, ksp, nbuffer;
    fd = open(fldt, O_RDWR | O_CREAT, 02644);
    if ( fd < 0 )
    {
        fprintf(stderr, "Can't open file %s\n", fldt);
        return(1);
    }
    temp = nh;
    wrtn = write(fd, &temp, sizeof(float));
    if ( wrtn != sizeof(float) )
    {
        fprintf(stderr, "write has failed at point 2.\n");
        close(fd);
        return(1);
    }
    temp = nv;
    wrtn = write(fd, &temp, sizeof(float));
    if ( wrtn != sizeof(float) )
    {
        fprintf(stderr, "write has failed at point 3.\n");
        close(fd);
        return(1);
    }
    temp = kim + rho * kx;
    wrtn = write(fd, &temp, sizeof(float));
    if ( wrtn != sizeof(float) )
    {
        fprintf(stderr, "write has failed at point 4.\n");
        close(fd);
        return(1);
    }
    temp = kjm + rho * ky;
    wrtn = write(fd, &temp, sizeof(float));
    if ( wrtn != sizeof(float) )
    {
        fprintf(stderr, "write has failed at point 5.\n");
        close(fd);
        return(1);
    }
    temp = kip + rho * kx;

```

```

    {
        for ( pixel = 0; pixel < ni; pixel++)
        {
            fg[k] = fg[k - ni]; k++;
        }
    }
}
}
}
}
}
cising(&imgin);
return(0);
}

float *rtgvv, *rtyvv, *rggvv, *rgxvv, *rxgvv, *rxvvv, *tggvv, *tgyvv,

initpr_d(dd, dn, ki, kj, nx, ny, ni, nj, mh, mv, nh, nv, lx, ky)
long *dd, *ki, *nx, *ni, *mh, nh, lx;
long *dn, *kj, *ny, *nj, *mv, nv, ky;
{
    double sqrt1, sqrt2, ceil(), sqrt();
    long ddv, kvv, nxv, niv, kgi = 0, ksi = 0, kgit2p1, ksit2p1, ktp;
    long dnv, kvj, nyv, nvj, kgj = 0, ksj = 0, kgit2p1, ksjt2p1, k;
    ddv = 1; sgmb = 1.0; thre = 2.0; scl = 1.0;
    dnv = 1; sgmq = 1.0; sgmw = 1.0;
    for ( k = 1; k < lvs; k++)
    {
        sgm = sgm * sqrt(msg); sgmg = sgm; sgms = SMTH * sgm; sgmb = sgmb * SGMB;
        sgd = sgd * sqrt(dsg); sgdg = sgd; sgds = SMTH * sgd; sgmq = sgmq * SGMQ;
        scl = scl * sqrt(two); rhx = rhx * sqrt(two * msg); sgmw = sgmw * SGMW;
        if ( ddv == dnv )
        {
            dnv = 2 * ddv; sqrt1 = 1.0;
            ktp = thre * 2.0 * ceil((double)(0.5 * sqrt1 * sgmg));
            if ( ktp > kgi ) kgi = ktp;
            ktp = thre * 2.0 * ceil((double)(0.5 * sqrt1 * sgdg));
            if ( ktp > kgj ) kgj = ktp;
            ktp = thre * 3.0 * ceil((double)(0.5 * sqrt1 * sgms));
            if ( ktp > ksi ) ksi = ktp;
            ktp = thre * 3.0 * ceil((double)(0.5 * sqrt1 * sgds));
            if ( ktp > ksj ) ksj = ktp;
        }
        else
        {
            ddv = dnv; sqrt2 = sqrt(two);
            ktp = thre * 2.0 * ceil((double)(0.5 * sqrt2 * sgmg));
            if ( ktp > kgi ) kgi = ktp;
            ktp = thre * 2.0 * ceil((double)(0.5 * sqrt2 * sgdg));
            if ( ktp > kgj ) kgj = ktp;
            ktp = thre * 3.0 * ceil((double)(0.5 * sqrt2 * sgms));
            if ( ktp > ksi ) ksi = ktp;
            ktp = thre * 3.0 * ceil((double)(0.5 * sqrt2 * sgds));
            if ( ktp > ksj ) ksj = ktp;
        }
    }
}

```

【図76】

```

}
kgit2p1 = 2 * kgi + 1; kgjt2p1 = 2 * kgj + 1;
ksit2p1 = 2 * ksi + 1; ksjt2p1 = 2 * ksj + 1;
kiv = ksi * dnv; *nh = (nh - 1) / dnv + 1;
kjb = ksj * dnv; *nv = (nv - 1) / dnv + 1;
nxv = tau * (nh - 1) + 2 * kx + 1; *kd = kiv; *nx = nxv;
nyv = tau * (nv - 1) + 2 * ky + 1; *kj = kjb; *ny = nyv;
niv = rho * (nxv - 1) + 2 * kiv + 1; *dd = ddv; *ni = niv;
njv = rho * (nyv - 1) + 2 * kjb + 1; *dn = dnv; *nj = njv;
nij = niv * njv + 1;
nhv = nh * nv + 1;
nxy = nxv * nyv + 1;
rgvtr = (float *) malloc((unsigned)(ksjt2p1 * sizeof(float)));
if (!rgvtr)
{
    fprintf(stderr, "memory requested for array rgvtr failed, msp = %d\n", msp);
    return(1);
} else msp += ksjt2p1 * sizeof(float);
rtyvtr = (float *) malloc((unsigned)(ksjt2p1 * sizeof(float)));
if (!rtyvtr)
{
    fprintf(stderr, "memory requested for array rtyvtr failed, msp = %d\n", msp);
    return(1);
} else msp += ksjt2p1 * sizeof(float);
rggvtr = (float *) malloc((unsigned)(ksit2p1 * sizeof(float)));
if (!rggvtr)
{
    fprintf(stderr, "memory requested for array rggvtr failed, msp = %d\n", msp);
    return(1);
} else msp += ksit2p1 * sizeof(float);
rgxvtr = (float *) malloc((unsigned)(ksit2p1 * sizeof(float)));
if (!rgxvtr)
{
    fprintf(stderr, "memory requested for array rgxvtr failed, msp = %d\n", msp);
    return(1);
} else msp += ksit2p1 * sizeof(float);
rxgvtr = (float *) malloc((unsigned)(kgit2p1 * sizeof(float)));
if (!rxgvtr)
{
    fprintf(stderr, "memory requested for array rxgvtr failed, msp = %d\n", msp);
    return(1);
} else msp += kgit2p1 * sizeof(float);
rxxvtr = (float *) malloc((unsigned)(kgit2p1 * sizeof(float)));
if (!rxxvtr)
{
    fprintf(stderr, "memory requested for array rxxvtr failed, msp = %d\n", msp);
    return(1);
} else msp += kgit2p1 * sizeof(float);
tggvtr = (float *) malloc((unsigned)(niv * sizeof(float)));
if (!tggvtr)
{
    fprintf(stderr, "memory requested for array tggvtr failed, msp = %d\n", msp);
    return(1);
} else msp += niv * sizeof(float);
tgyvtr = (float *) malloc((unsigned)(niv * sizeof(float)));

```

【図77】

```

if (!tgyvtr)
{
    fprintf(stderr, "memory requested for array tgyvtr failed, msp = %d\n", msp);
    return(1);
} else msp += niv * sizeof(float);
fprintf(stderr, " msp = %d; ", msp);
return(0);
}

setter(dd, dn, ki, kj, ni, nx, ny, fgg, ggg, ggx, ggy, gxg, gxx, gxy, gxs, gsy)
long dd, dn, ki, kj, ni, nx, ny;
float *fgg, *ggg, *ggx, *ggy, *gxg, *gxx, *gxy, *gsx, *gsy;
/*****
*/
/* Subroutine **setter** computes the values of the following */
/* image functions: ggg, ggx, ggy, gxg, gxx, gxy, ggs, gxs, by */
/* correlating the values of the initial image function cgg */
/* with the values of the following two-dimensional arrays: */
/* rgg, rgx, rgy, rxg, rxx, rxy, rgs, rxs. */
/*
/* The array, ggg, is related to the array, cgg, as follows: */
/* ggg[x + nx * y] corresponds to cgg[i[x] + ni * j[y]] where: */
/* i[x] = ki + rho * x, and j[y] = kj + rho * y, for x = 0, */
/* ..., nx - 1, and y = 0, ..., ny - 1. */
/*
/* The arrays ggx, ggy, gxg, gxx, gxy, ggs, gxs, are related */
/* to the array cgg in the same way as the array ggg is */
/* related to the array cgg. */
/*
/* The variables are defined as follows. */
/*
/* Images (each represented as a 1-dimensional array, in a row */
/* major order): */
/*
/* cgg[i + ni * j], i = 0, ..., ni - 1, j = 0, ..., nj - 1, is */
/* the initial image value at the pixel (kki - dki + i) on the */
/* line (kkj - dkj + j); */
/*
/* ggg[x + nx * y], x = 0, ..., nx - 1, y = 0, ..., ny - 1, is */
/* correlated image value at pixel i[x] = (ki + rho * x) on the */
/* line j[y] = (kj + rho * y) in the i, j coordinate system; */
/* the correlation function rgg is defined to be a modified */
/* Gaussian function. */
/*
/* gxg[x + nx * y], x = 0, ..., nx - 1, y = 0, ..., ny - 1, is */
/* correlated image value at pixel i[x] = (ki + rho * x) on the */
/* line j[y] = (kj + rho * y) in the i, j coordinate system; */
/* the correlation function rxg is defined to be a modified */
/* first-order partial with respect to variable x derivative of */
/* the Gaussian function. */
/*
/* ggx[x + nx * y], x = 0, ..., nx - 1, y = 0, ..., ny - 1, */
/* is the value of the first-order partial with respect to the */
/* variable x derivative of the function, ggg, where the system */
/* of coordinates (x,y) relates to the system of coordinates */

```

【図78】

```

/* (i,j) as: i[x] = (ki + rho * x), j[y] = (kj + rho * y); */
/*
/* ggy(x + nx * y), x = 0, ..., nx - 1, y = 0, ..., ny - 1, */
/* is the value of the first-order partial with respect to the */
/* variable y derivative of the image, ggg, where the system of */
/* coordinates (x,y) relates to the system of coordinates (i,j) */
/* as: i[x] = (ki + rho * x), j[y] = (kj + rho * y); */
/*
/* gxx(x + nx * y), x = 0, ..., nx - 1, y = 0, ..., ny - 1, */
/* is the value of the first-order partial with respect to the */
/* variable x derivative of the function, gxg, where the system */
/* of coordinates (x,y) relates to the system of coordinates */
/* (i,j) as: i[x] = (ki + rho * x), j[y] = (kj + rho * y); */
/*
/* gxy(x + nx * y), x = 0, ..., nx - 1, y = 0, ..., ny - 1, */
/* is the value of the first-order partial with respect to the */
/* variable y derivative of the image, gxg, where the system of */
/* coordinates (x,y) relates to the system of coordinates (i,j) */
/* as: i[x] = (ki + rho * x), j[y] = (kj + rho * y); */
/*
/* ggs[x + nx * y], x = 0, ..., nx - 1, y = 0, ..., ny - 1, */
/* is the value of the smoothed version of the function, ggg, */
/* at the pixel i[x] = (ki + rho * x), on the line */
/* j[y] = (kj + rho * y); */
/*
/* gxs[x + nx * y], x = 0, ..., nx - 1, y = 0, ..., ny - 1, */
/* is the value of the smoothed version of the function, gxg, */
/* at the pixel i[x] = (ki + rho * x), on the line */
/* j[y] = (kj + rho * y); */
/*
/* Resolution pyramid: */
/*
/* (dn[level], level = 1, ..., levels), where: dn[1] = 1, and */
/* for level = 2, ..., levels dn[level] is defined as follows: */
/* dn[level] = 2 * dd[level-1], if dn[level-1] = dd[level-1] */
/* dn[level] = dd[level-1], if dn[level-1] = 2 * dd[level-1]; */
/*
/* (dd[level], level = 1, ..., levels), where: dd[1] = 1, and */
/* for level = 2, ..., levels dd[level] is defined as follows: */
/* dd[level] = dn[level-1], if dn[level-1] = 2 * dd[level-1] */
/* dd[level] = 2 * dn[level-1], if dn[level-1] = dd[level-1]; */
/*
/* For each level = 1, ..., levels, the Gaussian parameters: */
/*
/* kgi[level] = thre * ceil(sgmrg) * dn[level], */
/* kgj[level] = thre * ceil(sgdg) * dn[level], */
/* ksi[level] = thre * ceil(sgms) * dn[level], */
/* ksj[level] = thre * ceil(sgds) * dn[level], */
/*
/* ki = ksi[levels]; kj = ksj[levels]; */
/*
/* ..... */
{
    long k, nitdd; register float *fg, *tg, *ty, *rg, *ry;
    float *g0, *g1, *g2, *g3, *g4, *g5, *s0, *s1;

```

【図79】

```

float *rg, *ry, *rgg, *rgx, *rgy, *rsx, *rsy, *rsz;
long i, ii, kgi, ksi, ngi, nrk, ifgidd, ifsjdd, kgi2p1, ksi2p1;
long j, jj, kgj, ksj, nsi, igg, ifgjdd, ifsjdd, kgj2p1, ksj2p1;
double omeg, rump, dmp, tmpxs, sqrt1, sqrt0, sgmg, sgmsi;
double omsq, tmsq, dsci, tmpys, sqrt2, ceil(), sgdgj, sgdsj;
nitdd = ni * dd;
if ( dd == dn )
{
    sqrt1 = 1.0;
    kgi = thre * 2.0 * ceil((double)(0.5 * sqrt1 * sgmg));
    kgj = thre * 2.0 * ceil((double)(0.5 * sqrt1 * sgdg));
    ksi = thre * 3.0 * ceil((double)(0.5 * sqrt1 * sgms));
    ksj = thre * 3.0 * ceil((double)(0.5 * sqrt1 * sgds));
}
else
{
    sqrt2 = sqrt(rwo);
    kgi = thre * 2.0 * ceil((double)(0.5 * sqrt2 * sgmg));
    kgj = thre * 2.0 * ceil((double)(0.5 * sqrt2 * sgdg));
    ksi = thre * 3.0 * ceil((double)(0.5 * sqrt2 * sgms));
    ksj = thre * 3.0 * ceil((double)(0.5 * sqrt2 * sgds));
}
omeg = 1.0 / 3.0; omsq = omeg * omeg;
dmp = rho * tau; dsci = dd / dmp;
sgmg = sgmg * scl / dmp;
sgdg = sgdg * scl / dmp;
sgms = sgms * scl / dmp;
sgds = sgds * scl / dmp;
rg = rgv; k = 0;
ry = ryv; rump = 0.0;
for ( j = -kgj; j <= kgj; j++ )
{
    tmpys = j * dsci / sgdgj;
    tmsq = tmpys * tmpys;
    dmp = -0.5 * tmsq / (1.0 - omsq * tmsq);
    rg[k] = exp(dmp); rump = rump + rg[k];
    tmsq = 1.0 - omsq * tmsq;
    ry[k] = ry[k] * tmpys / (sgdgj * tmsq * tmsq);
    k++;
}
rump = sqrt((double)isc) / rump; k = 0;
for ( j = -kgj; j <= kgj; j++ )
{
    rg[k] = rg[k] * rump;
    ry[k] = ry[k] * rump;
    k++;
}
rump = 0.0; k = 0;
rgg = rggv; rgx = rgxv;
rgy = rgyv; rxx = rxxv;
for ( i = -kgi; i <= kgi; i++ )
{
    tmpxs = i * dsci / sgmg;
    tmsq = tmpxs * tmpxs;
    dmp = -0.5 * tmsq / (1.0 - omsq * tmsq);

```

【図83】

```

    arsd[ksp].u0vf[k0h0] = arsd[ksp].uuvf[k0h0];
}
arsd[ksp].v0vf[0] = arsd[ksp].vvvf[0];
for ( itm = 0; itm < mxn; itm++ )
{
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        if ( ( ist = scidr2(dd, dn, kx, ky, nx, ny) ) != 0 )
        {
            fprintf(stderr, "Subroutine corres: scidr2 ");
            fprintf(stderr, "returns value equal to %d\n", ist);
            return(ist);
        }
    }
}
rf0 = 0.0;
for ( ksp = 0; ksp < nsp; ksp++ )
{
    if ( ( ist = setmx(dd, dn, nh, nv, &rf0) ) != 0 )
    {
        fprintf(stderr, "Subroutine corres: setmx ");
        fprintf(stderr, "returns value equal to %d\n", ist);
        return(ist);
    }
}
for ( k = kv0h0; k; k-- )
{
    arsd[ksp].udvf[mv0h0[k]] = 0.0;
}
arsd[ksp].v0vf[0] = 0.0;
}
fepsg = feps * rf0;
fprintf(stderr, "rf = %8.6f; rk = ", rf0 / (float) (nsp * kv0h0));
fprintf(stdout, "rf = %8.6f; rk = ", rf0 / (float) (nsp * kv0h0));
for ( its = 0; its < mxs; its++ )
{
    conjgr(&rkf, fepsg, mxl);
    fprintf(stderr, "%8.6f; ", rkf / (float) (nsp * kv0h0));
    fprintf(stdout, "%8.6f; ", rkf / (float) (nsp * kv0h0));
}
tmpuu = 0.0; tmpvv = 0.0;
for ( ksp = 0; ksp < nsp; ksp++ )
{
    for ( k = kv0h0; k; k-- )
    {
        k0h0 = mv0h0[k];
        arsd[ksp].uuvf[k0h0] = arsd[ksp].uuvf[k0h0]
            + stp * arsd[ksp].udvf[k0h0];
        tmpuu += arsd[ksp].uuvf[k0h0];
    }
    arsd[ksp].vvvf[0] = arsd[ksp].vvvf[0]
        + stp * arsd[ksp].v0vf[0];
    tmpvv += arsd[ksp].vvvf[0];
}
temp = nsp * kv0h0; tmpuu = tmpuu / temp;
temp = nsp; tmpvv = tmpvv / temp;

```

【図80】

```

rgg[k] = exp(dtmp); rtmp = rtmp + rgg[k];
tmsq = (1.0 - omsq * tmsq) * (1.0 - omsq * tmsq);
rgx[k] = rgg[k] * tmpxs / (sgmgi * tmsq);
rxg[k] = rgg[k] * tmpxs / sgmgi;
rxx[k] = rgg[k] * (tmpxs * tmpxs / tmsq - 1.0) / (sgmgi * sgmgi);
k++;
}
rtmp = sqrt((double)isc) / rtmp; k = 0;
for (i = -kgi; i <= kgi; i++)
{
    rgg[k] = rgg[k] * rtmp;
    rgx[k] = rgx[k] * rtmp;
    rxg[k] = rxg[k] * rtmp;
    rxx[k] = rxx[k] * rtmp;
    k++;
}
if (dn == dd) mrk = 0; else mrk = 1;
kgi2p1 = 2 * kgi + 1; kgj2p1 = 2 * kgj + 1;
ifgidd = ni * (kj - kgj * dd) + ki - kgi * dd; igg = 0;
ngi = rho * (nx - 1) + 2 * kgi + 1;
for (j = 0; j < ny; j += dd)
{
    ifgidd = ifgidd;
    tg = tggvtr; ty = tgyvtr;
    for (i = 0; i < ngi; i += dd)
    {
        fg = fgg + ifgidd;
        *tg = 0.0; rg = rgg;
        *ty = 0.0; ry = rgy;
        for (jj = 0; jj < kgj2p1; jj++)
        {
            *tg += *fg * *rg++;
            *ty += *fg * *ry++;
            fg += ni * dd;
        }
        ifgidd += dd; tg++; ty++;
    }
    ifgidd += ni * rho * dd; tg = tggvtr; ty = tgyvtr;
    g0 = ggg + igg; g1 = ggx + igg; g2 = ggy + igg;
    g3 = gxg + igg; g4 = gxx + igg; g5 = gxy + igg;
    for (i = 0; i < nx; i += dd)
    {
        if (mrk >= 0)
        {
            rgg = rggvtr; rgx = rgxvtr;
            rxg = rxgvtr; rxx = rxxvtr;
            *g0 = 0.0; *g1 = 0.0; *g2 = 0.0;
            *g3 = 0.0; *g4 = 0.0; *g5 = 0.0;
            for (ii = 0; ii < kgi2p1; ii++)
            {
                *g0 += *tg * *rgg;
                *g1 += *tg * *rgx;
                *g2 += *ty * *rgg;
                *g3 += *tg * *rxg;
                *g4 += *tg * *rxx;
            }
        }
    }
}

```

【図87】

```

wrtn = write(fd, &temp, sizeof(float));
if (wrtn != sizeof(float))
{
    fprintf(stderr, "write has failed at point 6.\n");
    close(fd);
    return(1);
}
temp = kjp + rho * ky;
wrtn = write(fd, &temp, sizeof(float));
if (wrtn != sizeof(float))
{
    fprintf(stderr, "write has failed at point 7.\n");
    close(fd);
    return(1);
}
temp = rho;
wrtn = write(fd, &temp, sizeof(float));
if (wrtn != sizeof(float))
{
    fprintf(stderr, "write has failed at point 9.\n");
    close(fd);
    return(1);
}
temp = tau;
wrtn = write(fd, &temp, sizeof(float));
if (wrtn != sizeof(float))
{
    fprintf(stderr, "write has failed at point 10.\n");
    close(fd);
    return(1);
}
temp = nsp;
wrtn = write(fd, &temp, sizeof(float));
if (wrtn != sizeof(float))
{
    fprintf(stderr, "write has failed at point 16.\n");
    close(fd);
    return(1);
}
for (ksp = 0; ksp < nsp; ksp++)
{
    temp = optf.dtm[ksp] - optf.dtc[ksp];
    wrtn = write(fd, &temp, sizeof(float));
    if (wrtn != sizeof(float))
    {
        fprintf(stderr, "write has failed at the point 18: ");
        fprintf(stderr, "temp = %f; wrtn = %d; ", temp, wrtn);
        fprintf(stderr, "ksp = %d;\n", ksp);
        close(fd);
        return(1);
    }
    temp = optf.drp[ksp] + optf.dtc[ksp];
    wrtn = write(fd, &temp, sizeof(float));
    if (wrtn != sizeof(float))
    {

```

【図82】

```

fg = fgg + ifsid;
*tg = 0.0; rg = rg;
*ty = 0.0; ry = ry;
for (jj = 0; jj < ksit2p1; jj++)
{
    *tg += *fg * *rg++;
    *ty += *fg * *ry++;
    fg += niudd;
}
ifsid += dd; tg++; ty++;
}
ifsidd += ni * rho * dd;
s0 = gsx + igg; tg = tggvtr;
s1 = gsy + igg; ty = tgyvtr;
for (i = 0; i < nx; i += dd)
{
    if (mrk >= 0)
    {
        rsg = rggvtr; *s0 = 0.0;
        rsx = rgxvtr; *s1 = 0.0;
        for (ii = 0; ii < ksit2p1; ii++)
        {
            *s0 += *tg++ * *rsx++;
            *s1 += *ty++ * *rsg++;
        }
        tg -= ksit2p1; ty -= ksit2p1;
    }
    mrk = -mrk;
    s0 += dd; tg += rho;
    s1 += dd; ty += rho;
}
igg += nx * dd;
}
}
corres(dd, dn, kx, ky, nx, ny, mh, mv, nh, nv, reps, feps, mxn, mxu, mxl, mxs)
long dd, dn, kx, ky, nx, ny, mh, mv, nh, nv, mxn, mxu, mxl, mxs;
float reps, feps;
{
    register long k, k0h0; long ksc, nsc, im, itu, its, iust;
    float rkf, rf0, sip = STEP, stl, std, fepsg, umpuu, umpvv, temp;
    sid = STPD;
    if ((itst = setin1(dd, dn, mh, mv, nh)) != 0)
    {
        fprintf(stderr, "Subroutine corres: setin1 ");
        fprintf(stderr, "returns value equal to %d.\n", itst);
        return(iust);
    }
    if (mxn < 1) return(0);
    for (ksp = 0; ksp < nsp; ksp++)
    {
        for (k = kv0h0; k; k--)
        {
            k0h0 = mv0h0[k];

```

【図89】

```

scd = 2.0 * dd * dd; scd = 1.0 / scd;
scn = dn * dn; scn = 1.0 / scn;
for (k = kv0h0; k; k--)
{
    dusq[mv0h0[k]] = 0.0;
}
for (k = kvph0; k; k--)
{
    tempu = uuvf[mvphm[k]] - uuvf[mvphp[k]];
    tempw = tempu * tempu * scd;
    dusq[mvphm[k]] += tempw;
    dusq[mvphp[k]] += tempw;
}
for (k = kv0hp; k; k--)
{
    tempu = uuvf[mv0hm[k]] - uuvf[mv0hp[k]];
    tempw = tempu * tempu * scn;
    dusq[mv0hm[k]] += tempw;
    dusq[mv0hp[k]] += tempw;
}
for (k = kvmhp; k; k--)
{
    tempu = uuvf[mvphm[k]] - uuvf[mvmp[hk]];
    tempw = tempu * tempu * scd;
    dusq[mvphm[k]] += tempw;
    dusq[mvmp[hk]] += tempw;
}
for (k = kvmh0; k; k--)
{
    tempu = uuvf[mvph0[k]] - uuvf[mvmh0[k]];
    tempw = tempu * tempu * scn;
    dusq[mvph0[k]] += tempw;
    dusq[mvmh0[k]] += tempw;
}
auu1 = arsd[ksp].auu1;
cuv1 = arsd[ksp].cuv1;
bvv1 = arsd[ksp].bvv1;
duu1 = arsd[ksp].duvf;
dvv1 = arsd[ksp].dvvf;
for (k = kv0h0; k; k--)
{
    m = mv0h0[k];
    auu1[m] = 0.0;
    cuv1[m] = 0.0;
    duu1[m] = 0.0;
    umpg2[m] = 0.0;
    umpg3[m] = 0.0;
    umpg4[m] = 0.0;
    umpg5[m] = 0.0;
}
bvv1[0] = 0.0;
dvv1[0] = 0.0;
dip = optf.dip[ksp];
dic = optf.dic[ksp];
dim = optf.dim[ksp];

```

【図88】

```

    fprintf(stderr, "write has failed at point 19.\n");
    close(fd);
    return(1);
}
k = 1; nbuffer = 4 * nh;
if ( nbuffer > BUFEL )
{
    fprintf(stderr, "The buffer is not large enough\n");
    close(fd);
    return(1);
}
for ( jy = 0; jy < nv; jy++ )
{
    i = 0;
    for ( ix = 0; ix < nh; ix++ )
    {
        buffer[i] = hv0h0[k]; i++;
        buffer[i] = vv0h0[k]; i++;
        buffer[i] = arsd[ksp].uuvf[k]; i++;
        buffer[i] = arsd[ksp].vvvf[0]; i++;
        k++;
    }
    wrtn = write(fd, buffer, nbuffer * sizeof(float));
    if ( wrtn != nbuffer * sizeof(float) )
    {
        fprintf(stderr, "write has failed at point 23.\n");
        close(fd);
        return(1);
    }
}
close(fd);
return(0);
}

setdr2(dd, dn, kx, ky, nx, ny)
long dd, dn, kx, ky, nx, ny;
{
    double temp, tmp1, tmp3, exp0; register long k, l, m;
    float *hcp, *vcp, *ggm, *gxm, *gym, *xgm, *xxm, *xym, *sxm, *sym, rhgcn, sod;
    float *hcm, *vcm, *ggp, *gxp, *gyp, *xgp, *xxp, *xyp, *sxp, *syp, rhxcn, scn;
    float *dx1p, *dylp, *dx2p, *dy2p, *duul, *auul, *cuvl, gugt, gusq, gvsq, shx;
    float *dx1m, *dylm, *dx2m, *dy2m, *dvvl, *bvvl, *bcn, gvgt, gugv, gtsq, shy;
    float temp1, tmpgm, tempu, tmpgt, tmpgx, tmpxx, tmpsx, *tmpg2, *tmpg4, tempw;
    float temp2, tmpgp, tempv, tmpxt, tmpgy, tmpxy, tmpsy, *tmpg3, *tmpg5, *dusq;
    float *uuvf, dx1pk, dx2pk, dylpk, dy2pk, drp, dic, dcm;
    float *vvvf, dx1mk, dx2mk, dylmk, dy2mk;
    long k11p, k12p, k21p, k22p, bn;
    long k11m, k12m, k21m, k22m, it;
    hcp = wrk00; vcp = wrk01; hcm = wrk02; vcm = wrk03;
    dx1p = wrk04; dylp = wrk05; dx2p = wrk06; dy2p = wrk07;
    dx1m = wrk08; dylm = wrk09; dx2m = wrk10; dy2m = wrk11;
    tmpg2 = wrk12; tmpg3 = wrk13; tmpg4 = wrk14; tmpg5 = wrk15;
    dusq = arsd[ksp].dusq; bcn = wrk16;
    uuvf = arsd[ksp].uuvf; vvvf = arsd[ksp].vvvf;

```

【図90】

```

for ( k = kv0h0; k; k-- )
{
    m = mv0h0[k]; bcn[m] = 1.0;
    hcp[m] = hv0h0[m] + optf.dhp[ksp] + (dtp + dte) * uvvf[m];
    hcm[m] = hv0h0[m] + optf.dhm[ksp] - (dum - dte) * uvvf[m];
    vcp[m] = vv0h0[m] + optf.dvp[ksp] + (dtp + dte) * vvvf[0];
    vcm[m] = vv0h0[m] + optf.dvm[ksp] - (dum - dte) * vvvf[0];
}
it = setin2(bcn,hcp,vcp,m11p,m12p,m21p,m22p,dx1p,
            dy1p,dx2p,dy2p,dd,dn,kx,ky,nx,ny);
if ( it != 0 ) return(it);
it = setin2(bcn,hcm,vcm,m11m,m12m,m21m,m22m,dx1m,
            dy1m,dx2m,dy2m,dd,da,kx,ky,nx,ny);
if ( it != 0 ) return(it);
for ( bn = 0; bn < bns; bn++ )
{
    ggm = optf.ggm[bn]; ggp = optf.ggp[bn];
    gxm = optf.gxm[bn]; gxp = optf.gxp[bn];
    gym = optf.gym[bn]; gyp = optf.gyp[bn];
    xgm = optf.xgm[bn]; xgp = optf.xgp[bn];
    xxm = optf.xxm[bn]; xxp = optf.xxp[bn];
    xym = optf.xym[bn]; xyp = optf.xyp[bn];
    sxm = optf.sxm[bn]; sxp = optf.sxp[bn];
    sym = optf.sym[bn]; syx = optf.syx[bn];
    for ( k = kv0h0; k; k-- )
    {
        m = mv0h0[k];
        k11p = m11p[m]; dx1pk = dx1p[m];
        k12p = m12p[m]; dx2pk = dx2p[m];
        k21p = m21p[m]; dy1pk = dy1p[m];
        k22p = m22p[m]; dy2pk = dy2p[m];
        k11m = m11m[m]; dx1mk = dx1m[m];
        k12m = m12m[m]; dx2mk = dx2m[m];
        k21m = m21m[m]; dy1mk = dy1m[m];
        k22m = m22m[m]; dy2mk = dy2m[m];
        temp1 = dx2pk * ggp[k11p] + dx1pk * ggp[k12p];
        temp2 = dx2pk * ggp[k21p] + dx1pk * ggp[k22p];
        tmpgpg = dy2pk * temp1 + dy1pk * temp2;
        temp1 = dx2mk * ggm[k11m] + dx1mk * ggm[k12m];
        temp2 = dx2mk * ggm[k21m] + dx1mk * ggm[k22m];
        tmpgmg = dy2mk * temp1 + dy1mk * temp2;
        tmpgt = tmpgpg - tmpgmg;
        temp1 = dx2pk * gxp[k11p] + dx1pk * gxp[k12p];
        temp2 = dx2pk * gxp[k21p] + dx1pk * gxp[k22p];
        tmpgpg = dy2pk * temp1 + dy1pk * temp2;
        temp1 = dx2mk * gxm[k11m] + dx1mk * gxm[k12m];
        temp2 = dx2mk * gxm[k21m] + dx1mk * gxm[k22m];
        tmpgmg = dy2mk * temp1 + dy1mk * temp2;
        tmpgmx = dtp * tmpgpg + dum * tmpgmg;
        temp1 = dx2pk * gyp[k11p] + dx1pk * gyp[k12p];
        temp2 = dx2pk * gyp[k21p] + dx1pk * gyp[k22p];
        tmpgpg = dy2pk * temp1 + dy1pk * temp2;
        temp1 = dx2mk * gym[k11m] + dx1mk * gym[k12m];
        temp2 = dx2mk * gym[k21m] + dx1mk * gym[k22m];
        tmpgmg = dy2mk * temp1 + dy1mk * temp2;
    }
}

```

【図94】

```

nxtidn = nx * dn;
xmin = -kx; xmax = nx - kx - 1;
ymin = -ky; ymax = ny - ky - 1;
(register long k, k0h0, *rm11, *rm12, *rm21, *rm22;
 register float *rxc, *rdx1, *rdx2;
 register float *ryc, *rdy1, *rdy2;
 rm11 = m11; rm12 = m12; rm21 = m21; rm22 = m22;
 rxc = xc; rdx1 = dx1; rdx2 = dx2;
 ryc = yc; rdy1 = dy1; rdy2 = dy2;
 for ( k = kv0h0; k; k-- )
 {
     k0h0 = mv0h0[k];
     if ( rxc[k0h0] < xmin ) rxc[k0h0] = xmin;
     else if ( rxc[k0h0] > xmax ) rxc[k0h0] = xmax;
     if ( ryc[k0h0] < ymin ) ryc[k0h0] = ymin;
     else if ( ryc[k0h0] > ymax ) ryc[k0h0] = ymax;
     xcr = rxc[k0h0] / dn;
     ycr = ryc[k0h0] / dn;
     mx = floor((double)xcr);
     my = floor((double)ycr);
     rdx1[k0h0] = xcr - mx;
     rdy1[k0h0] = ycr - my;
     rdx2[k0h0] = 1.0 - rdx1[k0h0];
     rdy2[k0h0] = 1.0 - rdy1[k0h0];
     mx = kx + dn * mx;
     my = ky + dn * my;
     rm11[k0h0] = my * nx + mx;
     rm12[k0h0] = rm11[k0h0];
     if ( dx1[k0h0] != 0.0 ) rm12[k0h0] = rm12[k0h0] + dn;
     rm21[k0h0] = rm11[k0h0];
     if ( dy1[k0h0] != 0.0 ) rm21[k0h0] = rm21[k0h0] + nxdn;
     rm22[k0h0] = rm11[k0h0];
     if ( dx1[k0h0] != 0.0 ) rm22[k0h0] = rm22[k0h0] + dn;
     if ( dy1[k0h0] != 0.0 ) rm22[k0h0] = rm22[k0h0] + nxdn;
 }
 return(0);
}
if ( dn == (dd + dd) )
{
    nxdp = nx * dd + dd; nxdm = nx * dd - dd;
    xmin = dd - kx; xmax = nx - dd - kx - 1;
    ymin = dd - ky; ymax = ny - dd - ky - 1;
    (register long k, k0h0, *rm11, *rm12, *rm21, *rm22;
     register float *rxc, *rdx1, *rdx2;
     register float *ryc, *rdy1, *rdy2;
     rm11 = m11; rm12 = m12; rm21 = m21; rm22 = m22;
     rxc = xc; rdx1 = dx1; rdx2 = dx2;
     ryc = yc; rdy1 = dy1; rdy2 = dy2;
     for ( k = kv0h0; k; k-- )
     {
         k0h0 = mv0h0[k];
         if ( rxc[k0h0] < xmin ) rxc[k0h0] = xmin;
         else if ( rxc[k0h0] > xmax ) rxc[k0h0] = xmax;
         if ( ryc[k0h0] < ymin ) ryc[k0h0] = ymin;
         else if ( ryc[k0h0] > ymax ) ryc[k0h0] = ymax;
     }
}

```

【図91】

```

tmpgy = dtp * tmpgp + dtm * tmpgm;
temp1 = dx2pk * xgp[k11p] + dx1pk * xgp[k12p];
temp2 = dx2pk * xgp[k21p] + dx1pk * xgp[k22p];
tmpgp = dy2pk * temp1 + dy1pk * temp2;
temp1 = dx2mk * xgm[k11m] + dx1mk * xgm[k12m];
temp2 = dx2mk * xgm[k21m] + dx1mk * xgm[k22m];
tmpgm = dy2mk * temp1 + dy1mk * temp2;
tmpxt = tmpgp - tmpgm;
temp1 = dx2pk * xxp[k11p] + dx1pk * xxp[k12p];
temp2 = dx2pk * xxp[k21p] + dx1pk * xxp[k22p];
tmpgp = dy2pk * temp1 + dy1pk * temp2;
temp1 = dx2mk * xxm[k11m] + dx1mk * xxm[k12m];
temp2 = dx2mk * xxm[k21m] + dx1mk * xxm[k22m];
tmpgm = dy2mk * temp1 + dy1mk * temp2;
tmpxx = dtp * tmpgp + dtm * tmpgm;
temp1 = dx2pk * xyp[k11p] + dx1pk * xyp[k12p];
temp2 = dx2pk * xyp[k21p] + dx1pk * xyp[k22p];
tmpgp = dy2pk * temp1 + dy1pk * temp2;
temp1 = dx2mk * xym[k11m] + dx1mk * xym[k12m];
temp2 = dx2mk * xym[k21m] + dx1mk * xym[k22m];
tmpgm = dy2mk * temp1 + dy1mk * temp2;
tmpxy = dtp * tmpgp + dtm * tmpgm;
rhgcn = rhg * bcn[m]; rhxcn = rhx * bcn[m];
gtsq = rhgcn * tmpgt + tmpgt + rhxcn * tmpxt + tmpxt;
gugt = rhgcn * tmpgx + tmpgx + rhxcn * tmpxx + tmpxx;
gvgt = rhgcn * tmpgy + tmpgy + rhxcn * tmpxy + tmpxy;
gusq = rhgcn * tmpgx + tmpgx + rhxcn * tmpxx + tmpxx;
gugv = rhgcn * tmpgx + tmpgx + rhxcn * tmpxx + tmpxy;
gvsq = rhgcn * tmpgy + tmpgy + rhxcn * tmpxy + tmpxy;
temp = (arsd[ksp].psq + arsd[ksp].qsq * arsd[ksp].dusq[m]) * gtsq;
tmp1 = arsd[ksp].r * exp((double)(-1.0 * temp));
tmp3 = arsd[ksp].r * exp((double)(-3.0 * temp));
duul[m] = gugt * tmp1;
dvvl[0] = gvgt * tmp1;
auul[m] = gusq * tmp3;
cuvl[m] = gugv * tmp3;
bvvl[0] = gvsq * tmp3;
temp1 = dx2pk * sxp[k11p] + dx1pk * sxp[k12p];
temp2 = dx2pk * sxp[k21p] + dx1pk * sxp[k22p];
tmpgp = dy2pk * temp1 + dy1pk * temp2;
temp1 = dx2mk * sxm[k11m] + dx1mk * sxm[k12m];
temp2 = dx2mk * sxm[k21m] + dx1mk * sxm[k22m];
tmpgm = dy2mk * temp1 + dy1mk * temp2;
tmpsx = dtp * tmpgp + dtm * tmpgm;
temp1 = dx2pk * syp[k11p] + dx1pk * syp[k12p];
temp2 = dx2pk * syp[k21p] + dx1pk * syp[k22p];
tmpgp = dy2pk * temp1 + dy1pk * temp2;
temp1 = dx2mk * sym[k11m] + dx1mk * sym[k12m];
temp2 = dx2mk * sym[k21m] + dx1mk * sym[k22m];
tmpgm = dy2mk * temp1 + dy1mk * temp2;
tmpsy = dtp * tmpgp + dtm * tmpgm;
temp = tmpsx + tmpsy; tmpg2[m] += 0.5 * temp * temp;
temp = tmpsx; tmpg3[m] += temp * temp;
temp = tmpsx - tmpsy; tmpg4[m] += 0.5 * temp * temp;
temp = tmpsy; tmpg5[m] += temp * temp;

```

【図96】

```

{
    j = j1; j1 += nhtdn;
    for (ix = 0; ix < mh; ix++)
    {
        mv0h0[i] = j;
        i++; j += dn;
    }
}
i = 1; j1 = 1;
for (jy = 1; jy < mv; jy++)
{
    j = j1; j1 += nhtdn;
    for (ix = 1; ix < mh; ix++)
    {
        mvmahm[i] = j;
        mvphp[i] = j + nhp1 * dd;
        i++; j += dn;
    }
}
i = 1; j1 = 1;
for (jy = 0; jy < mv; jy++)
{
    j = j1; j1 += nhtdn;
    for (ix = 1; ix < mh; ix++)
    {
        mv0hm[i] = j;
        mv0hp[i] = j + dn;
        i++; j += dn;
    }
}
i = 1; j1 = dn * nh + 1;
for (jy = 1; jy < mv; jy++)
{
    j = j1; j1 += nhtdn;
    for (ix = 1; ix < mh; ix++)
    {
        mvphm[i] = j;
        mvmph[i] = j - nhm1 * dd;
        i++; j += dn;
    }
}
i = 1; j1 = dn * nh + 1;
for (jy = 1; jy < mv; jy++)
{
    j = j1; j1 += nhtdn;
    for (ix = 0; ix < mh; ix++)
    {
        mvph0[i] = j;
        mvmh0[i] = j - nhtdn;
        i++; j += dn;
    }
}
if (dn == dd) return(0);
i = mv * mh + 1;
j1 = dd * nh + dd + 1;

```

【図92】

```

    }
  }
  for ( k = kvphp; k; k-- )
  {
    arsd[ksp].g2sq[mvphp[k]] = tmpg2[mvphp[k]] + tmpg2[mvmhm[k]];
  }
  for ( k = kv0hp; k; k-- )
  {
    arsd[ksp].g3sq[mv0hp[k]] = tmpg3[mv0hp[k]] + tmpg3[mv0hm[k]];
  }
  for ( k = kvmhp; k; k-- )
  {
    arsd[ksp].g4sq[mvmhp[k]] = tmpg4[mvmhp[k]] + tmpg4[mvphm[k]];
  }
  for ( k = kvmh0; k; k-- )
  {
    arsd[ksp].g5sq[mvmh0[k]] = tmpg5[mvmh0[k]] + tmpg5[mvph0[k]];
  }
  return(0);
}

```

```

setin2(bcn, hc, vc, m11, m12, m21, m22, dx1, dy1, dx2, dy2, dd, dn, kx, ky, nx, ny)
long *m11, *m12, *m21, *m22, dd, dn, kx, ky, nx, ny;
float *bcn, *hc, *vc, *dx1, *dy1, *dx2, *dy2;

```

```

/*
/*   This subroutine perform the following functions.
/*
/*   1. The vector field
/*
/*   { (hc[k],vc[k]) | k = 1, ..., kv0h0 }
/*
/*   is transformed into the vector field
/*
/*   { (xc[k],yc[k]) | k = 1, ..., kv0h0 }
/*
/*   using the relations
/*
/*   xc[k] = kx + tau * (hc[k] - 1.0); k = 1, ..., kv0h0,
/*
/*   yc[k] = ky + tau * (vc[k] - 1.0); k = 1, ..., kv0h0.
/*
/*   2. Integer valued fields
/*
/*   { m11[k], m12[k], m21[k], m22[k] | k = 1, ..., kv0h0 }
/*
/*   are determined such that the matrix
/*
/*           m21[k] m22[k]
/*           m11[k] m12[k]
/*
/*   forms a grid cell for every k = 1, ..., kv0h0.
/*
/*   3. Real valued fields
/*
/*

```

【図93】

```

/*      { dx1[k], dy1[k], dx2[k], dy2[k] | k = 1, ..., kv0h0 }      */
/*                                                                    */
/*      are determined such that the function value f(xc[k],yc[k])    */
/*      is a bilinear interpolation of its values at the grid points    */
/*                                                                    */
/*      f(m11[k]), f(m12[k]), f(m21[k]), f(m22[k])                    */
/*                                                                    */
/*      defined by the relations                                        */
/*                                                                    */
/*      f(xc[k],yc[k]) = dy2[k] * (dx2[k] * f(m11[k]) + dx1[k] * f(m12[k])) */
/*                                                                    */
/*      + dy1[k] * (dx2[k] * f(m21[k]) + dx1[k] * f(m22[k])).        */
/*                                                                    */
/*                                                                    */
{
double floor();
float thre, thri, xmint, xmaxt, ymint, ymaxt;
long mx, my, mxx, myy, nxtdm, nxdp, nxdn;
float xmin, xmax, ymin, ymax, *xc, *yc, xcr, ycr;
xc = dx1; yc = dy1;
thre = tau * dn; thri = 1.0 / thre;
xmin = -kx; xmint = xmin + thre;
xmax = nx - kx - 1; xmaxt = xmax - thre;
ymin = -ky; ymint = ymin + thre;
ymax = ny - ky - 1; ymaxt = ymax - thre;
register long k, k0h0;
register float *rxc, *rhc, *ryc, *ryc;
rxc = xc; rhc = hc; ryc = yc; rvc = vc;
for ( k = kv0h0; k; k-- )
{
k0h0 = mv0h0[k];
rxc[k0h0] = tau * (rhc[k0h0] - 1.0);
ryc[k0h0] = tau * (rvc[k0h0] - 1.0);
if ( rxc[k0h0] <= xmin )
bcn[k0h0] = 0.0;
else if ( rxc[k0h0] < xmint )
bcn[k0h0] = bcn[k0h0] + (rxc[k0h0] - xmin) * thri
- bcn[k0h0] * (rxc[k0h0] - xmin) * thri;
if ( rxc[k0h0] >= xmax )
bcn[k0h0] = 0.0;
else if ( rxc[k0h0] > xmaxt )
bcn[k0h0] = bcn[k0h0] + (xmax - rxc[k0h0]) * thri
- bcn[k0h0] * (xmax - rxc[k0h0]) * thri;
if ( ryc[k0h0] <= ymin )
bcn[k0h0] = 0.0;
else if ( ryc[k0h0] < ymint )
bcn[k0h0] = bcn[k0h0] + (ryc[k0h0] - ymin) * thri
- bcn[k0h0] * (ryc[k0h0] - ymin) * thri;
if ( ryc[k0h0] >= ymax )
bcn[k0h0] = 0.0;
else if ( ryc[k0h0] > ymaxt )
bcn[k0h0] = bcn[k0h0] + (ymax - ryc[k0h0]) * thri
- bcn[k0h0] * (ymax - ryc[k0h0]) * thri;
}
}
if ( dn == dd )
{

```

【図95】

```

xcr = (ryc[k0h0] + rxc[k0h0]) / dn;
ycr = (ryc[k0h0] - rxc[k0h0]) / dn;
mx = floor((double)xcr);
my = floor((double)ycr);
rdx1[k0h0] = xcr - mx;
rdy1[k0h0] = ycr - my;
rdx2[k0h0] = 1.0 - rdx1[k0h0];
rdy2[k0h0] = 1.0 - rdy1[k0h0];
mxx = lcx + dd * (mx - my);
myy = ky + dd * (mx + my);
rm11[k0h0] = myy * nx + mxx;
rm12[k0h0] = rm11[k0h0];
if ( dxi[k0h0] != 0.0 ) rm12[k0h0] = rm12[k0h0] + nxdp;
rm21[k0h0] = rm11[k0h0];
if ( dy1[k0h0] != 0.0 ) rm21[k0h0] = rm21[k0h0] + nxdm;
rm22[k0h0] = rm11[k0h0];
if ( dxi[k0h0] != 0.0 ) rm22[k0h0] = rm22[k0h0] + nxdp;
if ( dy1[k0h0] != 0.0 ) rm22[k0h0] = rm22[k0h0] + nxdm;
})
return(0);
}
return(1);
}

setin1(dd, dn, mh, mv, nh)
long dd, dn, mh, mv, nh;
{
    register long i, j, ix, jy, jl;
    long nh1dn, nhp1, nhm1;
    nh1dn = nh * dn;
    nhp1 = nh + 1;
    nhm1 = nh - 1;
    if ( dn == dd ) {
        kv0h0 = mv * mh;
        kvphp = (mv - 1) * (mh - 1);
        kv0hp = mv * (mh - 1);
        kvmh0 = (mv - 1) * mh;
        kvmh1 = (mv - 1) * (mh - 1);
        kv0hm = mv * (mh - 1);
        kvphm = (mv - 1) * (mh - 1);
        kvph0 = (mv - 1) * mh;
    } else if ( dn == (dd + dd) ) {
        kv0h0 = mv * mh + (mv - 1) * (mh - 1);
        kvphp = (mv - 1) * (mh - 1) + (mv - 1) * (mh - 1);
        kv0hp = mv * (mh - 1) + (mv - 1) * (mh - 2);
        kvmh0 = (mv - 1) * (mh - 1) + (mv - 1) * (mh - 1);
        kvmh1 = (mv - 1) * mh + (mv - 2) * (mh - 1);
        kv0hm = mv * (mh - 1) + (mv - 1) * (mh - 1);
        kvphm = (mv - 1) * (mh - 1) + (mv - 1) * (mh - 1);
        kvph0 = (mv - 1) * mh + (mv - 2) * (mh - 1);
    } else return(1);
    i = 1; jl = 1;
    for ( jy = 0; jy < mv; jy++ )

```

【図97】

```

for ( jy = 1; jy < mv; jy++ )
{
    j = jl; jl += nh1dn;
    for ( ix = 1; ix < mh; ix++ )
    {
        mv0h0[i] = j;
        i++; j += dn;
    }
}
i = (mv - 1) * (mh - 1) + 1;
jl = dd * nh + dd + 1;
for ( jy = 1; jy < mv; jy++ )
{
    j = jl; jl += nh1dn;
    for ( ix = 1; ix < mh; ix++ )
    {
        mvmh0[i] = j;
        mvphp[i] = j + nhp1 * dd;
        i++; j += dn;
    }
}
i = mv * (mh - 1) + 1;
jl = dd * nh + dd + 1;
for ( jy = 1; jy < mv; jy++ )
{
    j = jl; jl += nh1dn;
    for ( ix = 2; ix < mh; ix++ )
    {
        mv0hm[i] = j;
        mv0hp[i] = j + dn;
        i++; j += dn;
    }
}
i = (mv - 1) * (mh - 1) + 1;
jl = dd * nh + dd + 1;
for ( jy = 1; jy < mv; jy++ )
{
    j = jl; jl += nh1dn;
    for ( ix = 1; ix < mh; ix++ )
    {
        mvphm[i] = j;
        mvphp[i] = j - nhm1 * dd;
        i++; j += dn;
    }
}
i = (mv - 1) * mh + 1;
jl = (dn + dd) * nh + dd + 1;
for ( jy = 2; jy < mv; jy++ )
{
    j = jl; jl += nh1dn;
    for ( ix = 1; ix < mh; ix++ )
    {
        mvph0[i] = j;
        mvmh0[i] = j - nh1dn;
        i++; j += dn;
    }
}

```

【図98】

```

    }
    return(0);
}

conjgr(rknf, feps, kmax)
long kmax;
float *rknf, feps;
{
    register long k, k0h0; long kit, ksv, ksc, nsc;
    float alphak, betak, rezd, fnsp, teps;
    float *ru, *pu, *qu, *zu, *du, *ud, *ai;
    float *rv, *pv, *qv, *zv, *dv, *vd, *bi;
    long *m11, *m12, *m21, *m22;
    float *w11, *w12, *w21, *w22;
    double ak, akml = 1.0, bk, diemp, sqrt();
    diemp = 0.0;
    for ( ksp = 0; ksp < nsp; ksp++ )
    {
        ru = arsd[ksp].rucv; rv = arsd[ksp].rvcv;
        pu = arsd[ksp].pucv; pv = arsd[ksp].pvcv;
        qu = arsd[ksp].qucv; qv = arsd[ksp].qvcv;
        du = arsd[ksp].duvf; dv = arsd[ksp].dvvf;
        ud = arsd[ksp].udvf; vd = arsd[ksp].vdfv;
        for ( k = kv0h0; k; k-- )
        {
            k0h0 = mv0h0[k];
            ru[k0h0] = du[k0h0];
            pu[k0h0] = ud[k0h0];
        }
        rv[0] = dv[0];
        pv[0] = vd[0];
        matrvr();
        nsc = arsd[ksp].cns;
        if ( ( nsp > 1 ) && ( nsc > 0 ) )
        {
            for ( ksc = 0; ksc < nsc; ksc++ )
            {
                ksv = arsd[ksp].cnv[ksc];
                m11 = arsd[ksp].m11[ksc]; m12 = arsd[ksp].m12[ksc];
                m21 = arsd[ksp].m21[ksc]; m22 = arsd[ksp].m22[ksc];
                w11 = arsd[ksp].w11[ksc]; w12 = arsd[ksp].w12[ksc];
                w21 = arsd[ksp].w21[ksc]; w22 = arsd[ksp].w22[ksc];
                ud = arsd[ksv].udvf; vd = arsd[ksv].vdfv;
                for ( k = kv0h0; k; k-- )
                {
                    k0h0 = mv0h0[k];
                    qu[k0h0] = w11[k0h0] * ud[m11[k0h0]]
                        + w12[k0h0] * ud[m12[k0h0]]
                        + w21[k0h0] * ud[m21[k0h0]]
                        + w22[k0h0] * ud[m22[k0h0]];
                }
                qv[0] = kv0h0 * arsd[ksp].wmn[ksc] * vd[0];
            }
        }
    }
}

```

【図99】

```

ru = arsd[ksp].rucv; rv = arsd[ksp].rvcv;
qu = arsd[ksp].qucv; qv = arsd[ksp].qvcv;
zu = arsd[ksp].zucv; zv = arsd[ksp].zvcv;
ai = arsd[ksp].ainv; bi = arsd[ksp].binv;
for ( k = kv0h0; k; k-- )
{
    k0h0 = mv0h0[k];
    ru[k0h0] = ru[k0h0] - qu[k0h0];
    zu[k0h0] = ai[k0h0] * ru[k0h0];
    diemp = diemp + ru[k0h0] * ru[k0h0];
}
rv[0] = rv[0] - qv[0];
zv[0] = bi[0] * rv[0];
diemp = diemp + rv[0] * rv[0];
}
*rknf = sqrt(diemp); if ( *rknf <= feps ) return;
fnsp = nsp; teps = feps / fnsp;
for ( ksp = 0; ksp < nsp; ksp++ )
{
    ru = arsd[ksp].rucv; rv = arsd[ksp].rvcv;
    pu = arsd[ksp].pucv; pv = arsd[ksp].pvcv;
    qu = arsd[ksp].qucv; qv = arsd[ksp].qvcv;
    zu = arsd[ksp].zucv; zv = arsd[ksp].zvcv;
    ud = arsd[ksp].udvf; vd = arsd[ksp].vdfv;
    ai = arsd[ksp].ainv; bi = arsd[ksp].binv;
    for ( kit = 0; kit < kmax; kit++ )
    {
        diemp = 0.0;
        for ( k = kv0h0; k; k-- )
        {
            k0h0 = mv0h0[k];
            diemp = diemp + ru[k0h0] * ru[k0h0];
        }
        diemp = diemp + rv[0] * rv[0];
        rezd = sqrt(diemp); if ( rezd < teps ) break;
        ak = 0.0;
        for ( k = kv0h0; k; k-- )
        {
            k0h0 = mv0h0[k];
            ak = ak + ru[k0h0] * zu[k0h0];
        }
        ak = ak + rv[0] * zv[0];
        if ( kit > 0 )
        {
            betak = ak / akml;
            for ( k = kv0h0; k; k-- )
            {
                k0h0 = mv0h0[k];
                pu[k0h0] = pu[k0h0] * betak + zu[k0h0];
            }
            pv[0] = pv[0] * betak + zv[0];
        } else {
            for ( k = kv0h0; k; k-- )
            {
                k0h0 = mv0h0[k];
            }
        }
    }
}

```

【図100】

```

    pu[k0h0] = zu[k0h0];
  }
  pv[0] = zv[0];
}
mainvr();
bk = 0.0;
for ( k = kv0h0; k; k-- )
{
  k0h0 = mv0h0[k];
  bk = bk + pu[k0h0] * qu[k0h0];
}
bk = bk + pv[0] * qv[0];
alphak = bk / bk;
for ( k = kv0h0; k; k-- )
{
  k0h0 = mv0h0[k];
  ud[k0h0] += alphak * pu[k0h0];
  ru[k0h0] -= alphak * qu[k0h0];
  zu[k0h0] = ai[k0h0] * ru[k0h0];
}
vd[0] += alphak * pv[0];
rv[0] -= alphak * qv[0];
zv[0] = bi[0] * rv[0];
akml = ak;
}
}
)

setmtr(dd, dn, nh, nv, rkf)
long dd, dn, nh, nv;
float *rkf;
{
  register long k, m, kru1, km2;
  long s, k1, *m1, *m2, ist;
  double dtmp, dvtmp, dutdu, temp, tmp1, tmp3, sqri0, exp0;
  float *suma, *smud, *uuvf, *utvf, *duvf, *auu1, *bvv1, scn;
  float *sumb, *smvd, *vvvf, *vrvf, *dvvf, *cuv1, *ainv, *binv;
  float *gg, *auu, scsq, scs, scd;
  uuvf = arsd[ksp].uuvf; vvvf = arsd[ksp].vvvf;
  utvf = arsd[ksp].utvf; vrvf = arsd[ksp].vrvf;
  duvf = arsd[ksp].duvf; dvvf = arsd[ksp].dvvf;
  suma = arsd[ksp].rucv; sumb = arsd[ksp].rvcv;
  smud = arsd[ksp].zucv; smvd = arsd[ksp].zvcv;
  ainv = arsd[ksp].ainv; binv = arsd[ksp].binv;
  auu1 = arsd[ksp].auu1; bvv1 = arsd[ksp].bvv1;
  cuv1 = arsd[ksp].cuv1; sumb[0] = 0.0; smvd[0] = 0.0;
  for ( k = kv0h0; k; k-- )
  {
    m = mv0h0[k];
    suma[m] = gul;
    smud[m] = - gun;
    sumb[0] += gvl;
    smvd[0] -= gvn;
  }
  if ( ( ist = setvr(suma, sumb, smud, smvd, dd, dn, nh, nv) ) != 0 )

```

【図101】

```

{
    fprintf(stderr, "Subroutine setmtx: server ");
    fprintf(stderr, "returns value equal to %d\n", ist);
    return(ist);
}
scn = dn; scd = sqrt(two) * dd;
scn = 1.0 / scn; scd = 1.0 / scd;
for ( s = 2; s < 6; s++ )
{
    switch (s)
    {
    case 2:
        k1 = kvphp;
        m1 = mvmhmx;
        m2 = mvphp;
        gg = arsd[ksp].g2sq;
        auu = arsd[ksp].auu2;
        scs = scd;
        break;
    case 3:
        k1 = kv0hp;
        m1 = mv0hm;
        m2 = mv0hp;
        gg = arsd[ksp].g3sq;
        auu = arsd[ksp].auu3;
        scs = scn;
        break;
    case 4:
        k1 = kvmhp;
        m1 = mvphm;
        m2 = mvmhph;
        gg = arsd[ksp].g4sq;
        auu = arsd[ksp].auu4;
        scs = scd;
        break;
    case 5:
        k1 = kvmh0;
        m1 = mvph0;
        m2 = mvmh0;
        gg = arsd[ksp].g5sq;
        auu = arsd[ksp].auu5;
        scs = scn;
        break;
    }
    scssq = scs * scs;
    for ( k = k1; k-- )
    {
        km1 = m1[k]; km2 = m2[k];
        dutmp = scs * (uuvf[km2] - uuvf[km1]); dutdu = dutmp * dutmp;
        temp = (arsd[ksp].csq + arsd[ksp].bsq * gg[km2]) * dutdu;
        tmp1 = arsd[ksp].a * scssq * exp((double)(-1.0 * temp));
        tmp3 = arsd[ksp].a * scssq * exp((double)(-3.0 * temp));
        auu[km2] -= tmp3; suma[km1] += tmp3; suma[km2] += tmp3;
        duvf[km1] += uuvf[km2] * tmp1; smud[km1] -= tmp1;
        duvf[km2] += uuvf[km1] * tmp1; smud[km2] -= tmp1;
    }
}

```

【図102】

```

    }
    temp = 0.0;
    for ( k = kv0h0; k; k-- )
    {
        m = mv0h0[k];
        auu1[m] += suma[m];
        duvf[m] += uvvf[m] + smud[m] * uvvf[m];
        temp = temp + duvf[m] * duvf[m];
    }
    bvv1[0] += sumb[0];
    dvvf[0] += vrvf[0] + smvd[0] * vrvf[0];
    temp = temp + dvvf[0] * dvvf[0];
    *rtf = *rtf + sqrt(temp);
    for ( k = kv0h0; k; k-- )
    {
        m = mv0h0[k];
        ainu[m] = 1.0 / auu1[m];
    }
    binv[0] = 1.0 / bvv1[0];
    return(0);
}

mainvr()
{
    register long k; float *pu, *pv, *qu, *qv;
    float *av0h0, *avphp, *av0hp, *avmhp, *avmh0, *cv0h0;
    av0h0 = arsd[ksp].auu1; cv0h0 = arsd[ksp].cuv1;
    avphp = arsd[ksp].auu2; av0hp = arsd[ksp].auu3;
    avmhp = arsd[ksp].auu4; avmh0 = arsd[ksp].auu5;
    pu = arsd[ksp].pucv; pv = arsd[ksp].pvcv;
    qu = arsd[ksp].qecv; qv = arsd[ksp].qvcv;
    {register long k0h0;
    qv[0] = arsd[ksp].bvv1[0] * pv[0];
    for ( k = kv0h0; k; k-- )
    {
        k0h0 = mv0h0[k];
        qu[k0h0] = av0h0[k0h0] * pu[k0h0] + cv0h0[k0h0] * pv[0];
        qv[0] = qv[0] + cv0h0[k0h0] * pu[k0h0];
    }
    {register long kmhm, kphp;
    for ( k = kvphp; k; k-- )
    {
        kmhm = mvhm[k]; kphp = mvphp[k];
        qu[kmhm] = qu[kmhm] + avphp[kphp] * pu[kphp];
    }
    {register long k0hm, k0hp;
    for ( k = kv0hp; k; k-- )
    {
        k0hm = mv0hm[k]; k0hp = mv0hp[k];
        qu[k0hm] = qu[k0hm] + av0hp[k0hp] * pu[k0hp];
    }
    {register long kphm, kmhp;
    for ( k = kvmhp; k; k-- )
    {

```

【図105】

```

    if ( vc[ktmp] > fnv ) vc[ktmp] = fnv;
    }
    if ( dn == dd )
    {
        for ( k = kv0h0; k; k-- )
        {
            ktmp = mv0h0[k];
            rhs = hc[ktmp] / fdn; rhm = floor(rhs);
            rvs = vc[ktmp] / fdn; rvm = floor(rvs);
            dh1[ktmp] = rhs - rhm; dh2[ktmp] = 1.0 - dh1[ktmp];
            dv1[ktmp] = rvs - rvm; dv2[ktmp] = 1.0 - dv1[ktmp];
            rhm = rhm * dn; rvm = rvm * dn;
            rhp = rhm; if ( rhp < nhm1 ) rhp = rhp + dn;
            rvp = rvm; if ( rvp < nvm1 ) rvp = rvp + dn;
            m11[ktmp] = nh * rvm + rhm + 1;
            m12[ktmp] = nh * rvm + rhp + 1;
            m21[ktmp] = nh * rvp + rhm + 1;
            m22[ktmp] = nh * rvp + rhp + 1;
            if ((rhm < 0) || (rhm > nhm1) || (rhp < 0) || (rhp > nhm1))
            {
                fprintf(stderr, "hc[%d] = %f; ", ktmp, hc[ktmp]);
                fprintf(stderr, "dh1[%d] = %f; ", ktmp, dh1[ktmp]);
                fprintf(stderr, "dh2[%d] = %f; ", ktmp, dh2[ktmp]);
                fprintf(stderr, "rhm = %d; ", rhm);
                fprintf(stderr, "rhp = %d;\n", rhp);
                return(1);
            }
            if ((rvm < 0) || (rvm > nvm1) || (rvp < 0) || (rvp > nvm1))
            {
                fprintf(stderr, "vc[%d] = %f; ", ktmp, vc[ktmp]);
                fprintf(stderr, "dv1[%d] = %f; ", ktmp, dv1[ktmp]);
                fprintf(stderr, "dv2[%d] = %f; ", ktmp, dv2[ktmp]);
                fprintf(stderr, "rvm = %d; ", rvm);
                fprintf(stderr, "rvp = %d;\n", rvp);
                return(1);
            }
        }
        return(0);
    }
    if ( dn == (dd + dd) )
    {
        for ( k = kv0h0; k; k-- )
        {
            ktmp = mv0h0[k];
            rhs = (vc[ktmp] + hc[ktmp]) / fdn;
            rvs = (vc[ktmp] - hc[ktmp]) / fdn;
            rhm = floor(rhs); rvm = floor(rvs);
            dh1[ktmp] = rhs - rhm; dh2[ktmp] = 1.0 - dh1[ktmp];
            dv1[ktmp] = rvs - rvm; dv2[ktmp] = 1.0 - dv1[ktmp];
            rhp = rhm - rvm; rvp = rhm + rvm;
            rhm = rhp * dd; rvm = rvp * dd;
            tmp1 = nh * rvm + rhm + 1;
            tmp2 = nh * (rvm + dn) + rhm + 1;
            tmp3 = nh * (rvm + dd) + rhm + dd + 1;
            tmp4 = nh * (rvm + dd) + rhm - dd + 1;

```

【図103】

```

kphm = mvphm[k]; kmhp = mvmhk[k];
qu[kphm] = qu[kphm] + avmhk[kmhp] * pu[kmhp];
})
(register long kph0, kmh0;
for (k = kvmh0; k; k--)
{
kph0 = mvph0[k]; kmh0 = mvmh0[k];
qu[kph0] = qu[kph0] + avmh0[kmh0] * pu[kmh0];
})
(register long kmhm, kphp;
for (k = kvmh0; k; k--)
{
kmhm = mvmh0[k]; kphp = mvph0[k];
qu[kphp] = qu[kphp] + avph0[khm] * pu[khm];
})
(register long k0hm, k0hp;
for (k = kv0hm; k; k--)
{
k0hm = mv0hm[k]; k0hp = mv0hp[k];
qu[k0hp] = qu[k0hp] + av0hp[k0hm] * pu[k0hm];
})
(register long kphm, kmhp;
for (k = kvph0; k; k--)
{
kphm = mvph0[k]; kmhp = mvmh0[k];
qu[kmhp] = qu[kmhp] + avmh0[kmhp] * pu[kphm];
})
(register long kph0, kmh0;
for (k = kvph0; k; k--)
{
kph0 = mvph0[k]; kmh0 = mvmh0[k];
qu[kmh0] = qu[kmh0] + avmh0[kmh0] * pu[kph0];
})
)

setind(hc, vc, m11, m12, m21, m22, dh1, dv1, dh2, dv2, dd, dn, nh, nv)
long *m11, *m12, *m21, *m22, dd, dn, nh, nv;
float *hc, *vc, *dh1, *dv1, *dh2, *dv2;
/*****
/*
/* This subroutine perform the following functions. */
/*
/* 1. The input vector field */
/*
/* { (hc[mv0h0[k]],vc[mv0h0[k]]) | k = 1,..., kv0h0 } */
/*
/* is transformed into the current vector field */
/*
/* { (hc[mv0h0[k]],vc[mv0h0[k]]) | k = 1,..., kv0h0 } */
/*
/* using the relations */
/*
/* hc[mv0h0[k]] = max(1.0,min(nh,hc[mv0h0[k]])) - 1.0; */
/*
/* vc[mv0h0[k]] = max(1.0,min(nv,vc[mv0h0[k]])) - 1.0; */

```

【図104】

```

/*                                     */
/* for k = 1, ..., kv0h0.             */
/*                                     */
/* 2. Integer valued fields            */
/*                                     */
/* (m11[mv0h0[k]], m12[mv0h0[k]], m21[mv0h0[k]], m22[mv0h0[k]]), */
/* k = 1, ..., kv0h0 are determined such that the matrix */
/*                                     */
/*      m11[mv0h0[k]] m12[mv0h0[k]] */
/*      m21[mv0h0[k]] m22[mv0h0[k]] */
/*                                     */
/* forms a grid cell of size dn containing the vector */
/* (hc[mv0h0[k]], vc[mv0h0[k]]), for every k = 1, ..., kv0h0. */
/*                                     */
/* 3. Real valued fields               */
/*                                     */
/* (dx1[mv0h0[k]], dy1[mv0h0[k]], dx2[mv0h0[k]], dy2[mv0h0[k]]), */
/* k = 1, ..., kv0h0 are determined such that the function value */
/* f(xc[mv0h0[k]], yc[mv0h0[k]]) is a bilinear interpolation of */
/* its values at the grid points */
/*                                     */
/*      f(m11[mv0h0[k]]) f(m12[mv0h0[k]]) */
/*      f(m21[mv0h0[k]]) f(m22[mv0h0[k]]) */
/*                                     */
/* defined by the relations */
/*                                     */
/* f(xc[mv0h0[k]], yc[mv0h0[k]]) = */
/* dy2[mv0h0[k]] * ( dx2[mv0h0[k]] * f(m11[mv0h0[k]]) */
/*      + dx1[mv0h0[k]] * f(m12[mv0h0[k]]) ) + */
/* dy1[mv0h0[k]] * ( dx2[mv0h0[k]] * f(m21[mv0h0[k]]) */
/*      + dx1[mv0h0[k]] * f(m22[mv0h0[k]]) ). */
/*                                     */
/* *****/
{
    float fnh, fnv;
    register long k, ktmp;
    double floor0, rths, rvs, fdn;
    long tmp1, tmp3, rthm, rvrm, nhm1;
    long tmp2, tmp4, rthp, rvvp, nvml;
    nhm1 = nh - 1; nvml = nv - 1;
    fnh = nhm1; fnv = nvml; fdn = dn;
    for ( k = kv0h0; k; k-- )
    {
        ktmp = mv0h0[k];
        hc[ktmp] -= 1.0;
        if ( hc[ktmp] < 0.0 ) hc[ktmp] = 0.0;
        if ( hc[ktmp] > fnh ) hc[ktmp] = fnh;
        vc[ktmp] -= 1.0;
        if ( vc[ktmp] < 0.0 ) vc[ktmp] = 0.0;
    }
}

```

【図106】

```

if ( rvm == -dd )
    m11[ktmp] = tmp2;
else
    m11[ktmp] = tmp1;
if ( rvm == nvm1 )
    m12[ktmp] = tmp1;
else if ( rhm == nhm1 )
    m12[ktmp] = tmp4;
else
    m12[ktmp] = tmp3;
if ( rvm == nvm1 )
    m21[ktmp] = tmp1;
else if ( rhm == 0 )
    m21[ktmp] = tmp3;
else
    m21[ktmp] = tmp4;
if ( rvm >= nvm1 - dd )
    m22[ktmp] = tmp1;
else
    m22[ktmp] = tmp2;
if ( ( rhm < 0 ) || ( rhm > nhm1 ) )
{
    fprintf(stderr, "ch: dd = %d; ", dd);
    fprintf(stderr, "nhm1 = %d; ", nhm1);
    fprintf(stderr, "rhm = %d;\n", rhm);
    return(1);
}
if ( ( rvm + dd < 0 ) || ( rvm > nvm1 ) )
{
    fprintf(stderr, "cv: dd = %d; ", dd);
    fprintf(stderr, "nvm1 = %d; ", nvm1);
    fprintf(stderr, "rvm = %d;\n", rvm);
    return(1);
}
}
return(0);
}
return(1);
}

servtr(suma, sumb, smud, smvd, dd, dn, nh, nv)
float *suma, *sumb, *smud, *smvd;
long dd, dn, nh, nv;
{
    register long l, k, ktmp; double dtmp, exp0;
    long ist, nsc, ksc, ksv, *m11, *m12, *m21, *m22;
    float tmpu, tu00, tu11, tu12, tu21, tu22, tw11, tw21, tw12, tw22, one = 1.5;
    float *utvf, *uuvf, *hc, *dh1, *dh2, *w11, *w12, *w00, temp, dlt, wxy, nhf;
    float *vtvf, *vvvf, *vc, *dv1, *dv2, *w21, *w22, *dusq, wmn, wuv, nvf;
    utvf = arsd[ksp].utvf; hc = wrk05; nhf = nh - 0.5;
    vtvf = arsd[ksp].vtvf; vc = wrk06; nvf = nv - 0.5;
    for ( k = kv0h0; k; k-- )
    {
        ktmp = mv0h0[k];
        utvf[ktmp] = gun * arsd[ksp].u0vf[ktmp];
    }
}

```

【図 107】

```

    }
    vrvf[0] = kv0h0 * gvn * arsd[ksp].v0vf[0];
    nsc = arsd[ksp].cns;
    if ( ( nsp < 2 ) || ( nsc < 1 ) ) return(0);
    for ( ksc = 0; ksc < nsc; ksc++)
    {
        wmn = arsd[ksp].wmn[ksc];
        wxy = arsd[ksp].wxy[ksc] * scln * sgmw;
        wuv = arsd[ksp].wuv[ksc] * scln * sgmw;
        dlt = arsd[ksp].dlt[ksc]; ksv = arsd[ksp].cns[ksc];
        dh1 = arsd[ksp].w11[ksc]; dh2 = arsd[ksp].w12[ksc];
        dv1 = arsd[ksp].w21[ksc]; dv2 = arsd[ksp].w22[ksc];
        w11 = arsd[ksp].w11[ksc]; w12 = arsd[ksp].w12[ksc];
        w21 = arsd[ksp].w21[ksc]; w22 = arsd[ksp].w22[ksc];
        m11 = arsd[ksp].m11[ksc]; m12 = arsd[ksp].m12[ksc];
        m21 = arsd[ksp].m21[ksc]; m22 = arsd[ksp].m22[ksc];
        uuvf = arsd[ksp].uuvf; vvvf = arsd[ksp].vvvf;
        for ( k = kv0h0; k; k-- )
        {
            ktmp = mv0h0[k];
            hc[ktmp] = hv0h0[ktmp] + dlt * uuvf[ktmp];
            vc[ktmp] = vv0h0[ktmp] + dlt * vvvf[0];
        }
        if(((itst = setind(hc,vc,m11,m12,m21,m22,dh1,dv1,dh2,dv2,dd,dn,nh,nv)) != 0)
        {
            fprintf(stderr,"Subroutine setvtr: setind ");
            fprintf(stderr,"returns value equal to %d.\n", itst);
            return(itst);
        }
        dusq = arsd[ksv].dusq;
        for ( k = kv0h0; k; k-- )
        {
            ktmp = mv0h0[k]; tu00 = arsd[ksp].uuvf[ktmp];
            tu11 = arsd[ksv].uuvf[m11[ktmp]]; tmpu = tu11 - tu00;
            dtmp = (- wxy - wuv * dusq[m11[ktmp]]) * tmpu * tmpu;
            rw11 = wmn * dv2[ktmp] * dh2[ktmp] * exp(dtmp);
            tu12 = arsd[ksv].uuvf[m12[ktmp]]; tmpu = tu12 - tu00;
            dtmp = (- wxy - wuv * dusq[m12[ktmp]]) * tmpu * tmpu;
            rw12 = wmn * dv2[ktmp] * dh1[ktmp] * exp(dtmp);
            tu21 = arsd[ksv].uuvf[m21[ktmp]]; tmpu = tu21 - tu00;
            dtmp = (- wxy - wuv * dusq[m21[ktmp]]) * tmpu * tmpu;
            rw21 = wmn * dv1[ktmp] * dh2[ktmp] * exp(dtmp);
            tu22 = arsd[ksv].uuvf[m22[ktmp]]; tmpu = tu22 - tu00;
            dtmp = (- wxy - wuv * dusq[m22[ktmp]]) * tmpu * tmpu;
            rw22 = wmn * dv1[ktmp] * dh1[ktmp] * exp(dtmp);
            if ( (hv0h0[ktmp] < one) || (hv0h0[ktmp] > nhf)
                || (vv0h0[ktmp] < one) || (vv0h0[ktmp] > nvf) )
            {
                rw11 = wmn * dv2[ktmp] * dh2[ktmp];
                rw12 = wmn * dv2[ktmp] * dh1[ktmp];
                rw21 = wmn * dv1[ktmp] * dh2[ktmp];
                rw22 = wmn * dv1[ktmp] * dh1[ktmp];
            }
            temp = 0.0; tmpu = 0.0;
            w11[ktmp] = rw11; temp += rw11; tmpu += tu11 * rw11;
        }
    }

```

フロントページの続き

(51)Int. Cl.<sup>6</sup>

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 T 15/00

// H 0 4 N 1/387

9365-5L

G 0 6 F 15/72

4 5 0 A